



MicroBE

Etude d'un système de comptage d'eau

Rapport de stage de Valentin Mercenaro

Professeur référent : M. Valentin Gies

Maître de stage : M. Philippe Graille

# Remerciement

Je tiens à remercier toutes les personnes qui m'ont aidé et accompagnée à obtenir ce stage ainsi qu'à le mener à bien.

Je voudrais remercier dans l'entreprise qui m'a accueilli, MicroBE :

- M. Eric VIEUX COMBE, de m'avoir fait confiance et a accepté de m'accueillir dans son entreprise,
- M. Philippe GRAILLE, mon tuteur de stage qui a accepté de me prendre qui a eu la patience de m'aider, corriger mes erreurs et répondre à nombre de mes questions,
- Mme. Patricia SORDI, pour son aide apporter que cela soit avant ou pendant le stage,
- M. Frédéric DAVIGNON, pour sa bonne humeur quotidienne et les moments de travaux partagés,
- M. Alain LADURELLE, également pour sa bonne humeur et ses recettes de cuisine,
- M. Bernard BUZURU, de m'avoir aidé durant certaines étapes du projet

## Table des matières

Remerciement .....	0
Table des illustrations .....	2
Intro .....	3
L'entreprise .....	4
Secteur d'activité .....	4
Secteur d'activité général : .....	4
Spécificités de l'entreprise : .....	4
Le projet .....	5
Contexte .....	5
Objectif .....	5
Environnement de travail .....	6
Contraintes .....	7
Présentation du déroulé du suivi de projet .....	7
Activité .....	7
Etude de la turbine et de la charge de la super capa et son chargeur .....	9
Etude de la mesure de débit .....	15
Etude de l'afficheur .....	19
Missions réalisés extérieur au projet .....	22
Recherche des composants .....	24
Etude du microcontrôleur NXP .....	26
Conception du PCB .....	31
Bilan .....	31
Présentation des résultats .....	31
Prolongement du projet .....	34
Apport personnel .....	35
Conclusion .....	36
Lexique .....	37
Table des annexes .....	39
annexes .....	40

## Table des illustrations

Figure 1. Organigramme de l'entreprise .....	5
Figure 2 Poste de travail .....	7
Figure 3 Plan du système à réaliser .....	8
Figure 4 Schéma de câblage du chargeur .....	15
Figure 5 Intérieur de turbine .....	16
Figure 6 Installation pour simulation en condition presque réelle .....	17
Figure 7 débit en fonction de la fréquence .....	17
Figure 8 Schéma de câblage du système finale .....	18
Figure 9 Equation de conversion en débit.....	20
Figure 10 Affichage du terminale série .....	21
Figure 11 affichage écran e-paper .....	22
Figure 12 Câble d'antenne .....	24
Figure 13 Schéma des dépendances du code de l'écran .....	29
Figure 14 Schéma de câblage NXP .....	30

## Intro

Dans le cadre de ma formation de BUT3 GEII, spécialité électronique et systèmes embarqués, j'ai réalisé mon stage en entreprise au sein du bureau d'études MicroBE pour une durée de 14 semaines, une très petite entreprise (TPE) de moins d'une dizaine d'employés.

Ayant pour objectif de continuer mes études en école d'ingénieur, j'ai orienté ma recherche de stage vers un poste se rapprochant des responsabilités et des compétences attendues à un niveau ingénieur, tant sur le plan technique que sur la gestion de projet. Durant l'entretien avec l'entreprise il m'a été demandé sur quoi je préférerais travailler durant mon stage, dans quel domaine, afin de trouver un projet correspondant à mes attentes.

Suite à mes réponses MicroBE m'a confié comme mission « comment réaliser une carte électronique de mesure de consommation d'eau autonome ? » un sujet me permettant de travailler sur de la conception et du développement de projets dans un environnement technique mêlant conception électronique, programmation embarquée, touchant à la fois au software et au hardware

Afin de contextualiser cette expérience, durant ce rapport je vais expliquer ce qu'est MicroBE, sa place sur le marché, son secteur d'activité et spécificité. Ensuite sera développé la partie projet, son contexte, l'état dans lequel il était à mon arrivée, ce qu'il restait à faire dessus, le matériel qui était à ma disposition pour mener à bien ma mission, les problèmes et contraintes rencontrées, ce que j'ai réalisé durant ces 14 semaines et enfin l'état du projet à la fin de mon stage.

## L'entreprise

### Secteur d'activité

#### Secteur d'activité général :

L'entreprise microBE évolue dans le secteur des technologies de contrôle d'accès, un marché dynamique en constante évolution. Grâce aux technologies sans contact, notamment les badges RFID et les cartes à puce, les entreprises publiques et privées ainsi que les particuliers recherchent des solutions toujours plus sûres et connectées. Ce secteur répond à des besoins très variés comme la sécurisation d'espaces, la régulation d'accès, la gestion de présence, etc., et touche de nombreux domaines comme les entreprises, l'industrie, les ports, l'hôtellerie ou encore les centres de loisirs.

#### Spécificités de l'entreprise :

Créée en 1986 par Serge Ternoir et Bernard Buzuru, MicroBE s'est d'abord positionné comme un bureau d'études en électronique et informatique. Elle conçoit et développe des produits innovants sur cahier des charges ou selon les idées issues de son équipe. Très tôt, elle a su maîtriser la chaîne complète de développement, de la conception de prototypes à la fabrication et à la maintenance. MicroBE a toujours su rester proche de ses clients, en offrant des solutions sur mesure, fiables et évolutives.

Au fil des années, l'entreprise s'est impliquée dans plusieurs projets innovants, notamment dans les domaines médicaux, scolaires (gestion des accès en établissements comme les cartes au collège lycée Maintenon), et portuaires (gestion multiservices par badge RFID). Sa filiale Insilio, créée en 2005, vient renforcer cette spécialisation dans la gestion d'accès par carte à puce dans des environnements variés.

En 2024, MicroBE et Insilio ont été rachetées par le groupe EKHO Solutions. Cette acquisition marque une nouvelle phase de développement pour MicroBE. L'entreprise, dont la zone d'influence s'étend sur toute la France, propose une offre complète de contrôle d'accès : lecteurs, badges, logiciels, accompagnement à l'installation, formation des utilisateurs et contrats de maintenance. MicroBE reste un acteur de proximité et d'innovation, reconnu pour la qualité de ses solutions RFID adaptées à chaque besoin métier.

L'entreprise étant une TPE, elle est installée dans de petits locaux qui favorisent un environnement de travail convivial et une certaine proximité entre les membres de l'équipe. Bien que chacun dispose de son propre bureau, l'ambiance reste chaleureuse et les échanges sont fluides. La taille réduite de la structure implique un effectif restreint, ce qui confère à chaque poste une importance particulière.

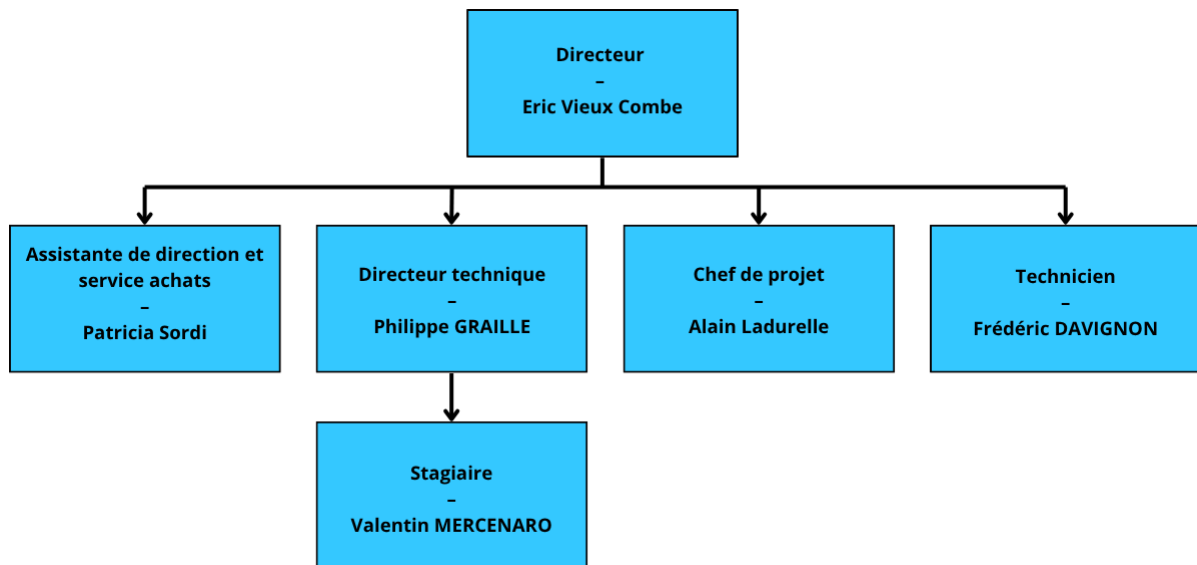


Figure 1. Organigramme de l'entreprise

## Le projet

### Contexte

L'entreprise a récemment été rachetée pour l'un de ses projets innovants : le développement d'un produit encore inexistant sur le marché. Toutefois, en raison d'un manque de ressources humaines, ce projet n'avait jusqu'alors pas connu d'avancée significative.

La mission qui m'a été confiée était donc de faire progresser au maximum le développement. Il s'agissait de poser les bases techniques du produit, d'explorer plusieurs pistes de conception et, si possible, de livrer une première version fonctionnelle. À défaut, au moins structurer suffisamment le projet pour en faciliter la poursuite après mon départ.

Le produit étant encore confidentiel, sa nature ne sera pas détaillée. Il s'agit d'un système électronique embarqué autonome destiné, entre autres, à la mesure du débit et de la consommation d'eau.

### Objectif

L'objectif du projet est de concevoir un système électronique embarqué totalement autonome en énergie. Ce dispositif repose sur une turbine hydraulique qui joue un double rôle, elle doit permettre de mesurer le débit d'eau mais aussi de produire l'énergie nécessaire au fonctionnement de l'ensemble du système. L'énergie générée est stockée dans une supercapacité (ou supercapa), laquelle alimente ensuite les différents composants électroniques.

Le cœur du système est un microcontrôleur à très basse consommation, chargé de piloter plusieurs modules : un capteur de température, un écran e-paper pour l'affichage local des

données, et un module de communication radio fonctionnant sur la bande des 868 MHz pour la transmission des informations à distance. L'écran e-paper a été choisi pour sa très faible consommation énergétique et sa capacité à conserver l'image même sans alimentation.

Le système doit également être capable de basculer en mode veille profonde pendant les longues périodes d'inactivité afin de garantir une autonomie pouvant aller de deux à sept jours. La logique de fonctionnement repose sur une consommation minimale en veille, combinée à une recharge plus rapide que la décharge lors des phases actives.

Pour garantir une interface simple, étanche et sans contact, des interrupteurs à lame souple (ILS) pourront être utilisés. Ces capteurs réagissent à un champ magnétique, ce qui permet de simuler l'appui sur un bouton sans aucun élément mécanique saillant, rendant le système plus résistant à l'humidité et aux conditions extérieures.

Enfin, l'ensemble du circuit doit être suffisamment compact pour être intégré dans un boîtier étanche de petite taille. Ce boîtier devra protéger les composants tout en permettant un usage en conditions réelles. Ce projet vise donc à réunir plusieurs défis techniques : autonomie énergétique, miniaturisation, robustesse environnementale et communication sans fil.

## Environnement de travail

Un poste de travail équipé d'un ordinateur a été mis à disposition dès le début du stage. Pour le bon déroulement du projet, l'ensemble du matériel de base requis a été fourni : turbine, écran e-paper, capteur de température, supercondensateurs ainsi que leur chargeur.

L'accès au matériel électronique disponible sur site était complet, incluant à la fois les composants divers et les instruments de mesure nécessaires (oscilloscope, multimètre, etc.), ainsi que les outils indispensables à l'assemblage et à la manipulation (fils, pinces à dénuder, fer à souder, pinces à sertir, etc.).

En cas de besoin, il était possible de commander du matériel ou des composants supplémentaires afin de répondre aux exigences techniques du projet. L'espace autour du bureau de travail pouvait être librement utilisé, ce qui a permis une organisation optimale pour la réalisation des montages et des phases de test dans de bonnes conditions.

Plusieurs logiciels ont également été utilisés tout au long du stage, comme Excel, Arduino, Visual Studio, Picoscope, MCUXpresso ou encore DesignSpark. Certains étaient déjà familiers, mais d'autres comme Picoscope, MCUXpresso et DesignSpark, que je découvrais, ont demandé un certain temps d'adaptation avant de pouvoir être utilisés efficacement dans le cadre du projet.



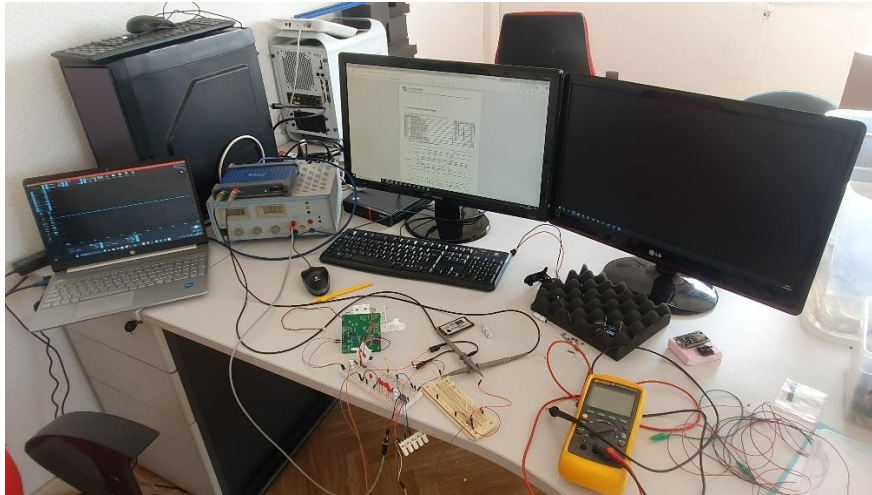


Figure 2 Poste de travail

## Contraintes

À mon arrivée, le projet était à un stade très peu avancé, des expériences avaient été réalisées notamment pour tester le bon fonctionnement des turbines et de la possibilité de mesure de débit sur celles-ci, vérifiée à l'aide d'un débitmètre. Cependant le matériel mis à ma disposition n'était pas le même que celui utilisé lors des précédents tests à part pour la turbine ce qui m'a conduit à reprendre le développement depuis zéro.

La majorité des composants utilisés, comme la turbine, l'écran e-paper ou encore la supercapacité, m'étaient inconnus, car des composants non étudiés durant le BUT. Leur prise en main a nécessité un temps d'apprentissage, accompagnée de recherches et de tests. À cela s'ajoutait un manque de matériel adapté, comme des câbles avec une utilisation spécifique que j'ai dû fabriquer moi-même car aucun de ceux à ma disposition ne répondait à mes attentes.

En parallèle, le projet reposait sur l'utilisation de composants peu coûteux d'origine chinoise sans trop d'informations ou même de datasheet disponible sur internet. Cela a rendu l'intégration plus complexe, nécessitant de nombreux tests pour comprendre leur fonctionnement.

Enfin, le microcontrôleur final prévu fonctionnait sous un environnement de développement NXP en langage C de bas niveau, que je ne connaissais pas au début du stage. Mes expériences antérieures portaient principalement sur MPLAB, Arduino et Visual Studio. Cette nouveauté a donc demandé une phase d'adaptation supplémentaire.

## Présentation du déroulé du suivi de projet

### Activité

Dans le cadre de ce projet, une organisation rigoureuse était essentielle afin d'assurer une progression structurée et réaliste. Bien que l'objectif final ait été clairement défini dès le départ,

le chemin pour y parvenir restait à identifier. Il a donc été nécessaire de découper le projet en plusieurs étapes successives, avec pour chacune une deadline à essayer de respecter. L'objectif de cette planification était de maximiser l'avancement du projet durant la période de stage, en se donnant une visibilité sur les priorités et le temps disponible.

Un document interne fourni en début de mission listait les composants à intégrer sur la future carte électronique. Certains éléments y figuraient comme indispensables, d'autres comme optionnels. L'attention a donc été portée en priorité sur les composants obligatoires : la turbine, le circuit de charge de la supercapacité, l'afficheur, le capteur de température et le module de communication radio. À partir de cette base, plusieurs grandes étapes techniques ont été identifiées :

- Étude de la turbine et du système de charge de la supercapacité,
- Étude de la mesure de débit,
- Intégration et pilotage de l'afficheur,
- Étude du microcontrôleur NXP

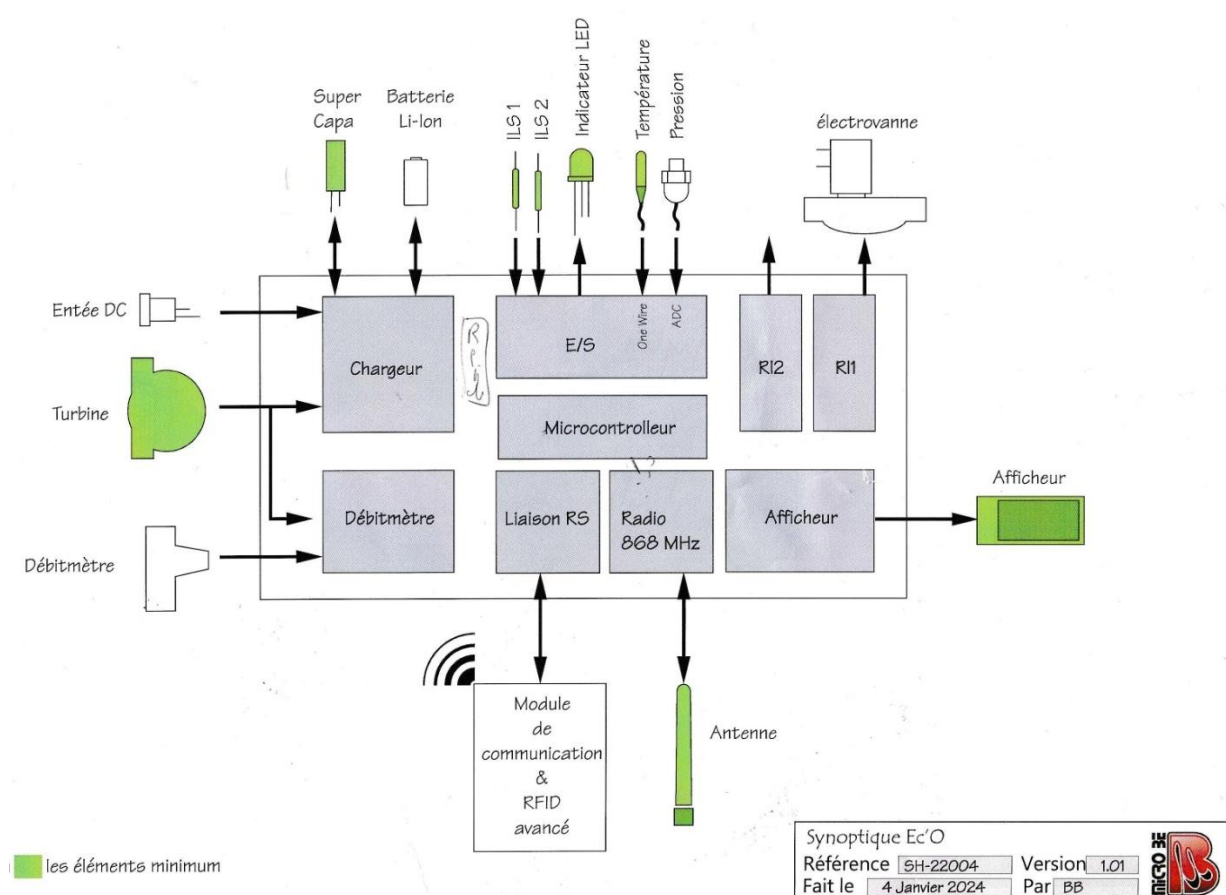


Figure 3 Plan du système à réaliser

Un planning prévisionnel a été établi (voir annexe 1) permettant de répartir les différentes tâches tout au long de la période de stage. Toutefois, la durée réelle de chaque étape a parfois fortement

divergé de ce qui était prévu. Certaines phases se sont révélées bien plus rapides que prévu tandis que d'autres ont nécessité davantage de temps en raison de difficultés techniques ou de contraintes matérielles. Certaines parties du projet n'ont pas pu être entamées ou finalisées dans le temps imparti. Cette gestion du planning a permis d'adapter la charge de travail en fonction des aléas techniques rencontrés tout en conservant une ligne directrice cohérente.

### *Etude de la turbine et de la charge de la super capa et son chargeur*

L'étude de la turbine s'est déroulée en deux phases. Dans un premier temps, des tests ont été réalisés à l'aide d'un moteur entraînant directement la turbine, afin de simuler les conditions de fonctionnement du système.

La turbine est capable de générer une tension comprise entre 0 et 80 V. Avec le moteur utilisé lors des essais, la tension maximale obtenue est inférieure à 80 V, mais reste, théoriquement, suffisante pour permettre la recharge d'une supercapacité.

La supercapacité est connectée à un circuit, un PCB appelé chargeur de supercapacité, dont le rôle est d'assurer une charge sécurisée et contrôlée. Ce module accepte une tension d'entrée comprise entre 0 et 40 V. Cependant, au-delà de 18 V, le chargeur se met en sécurité bloquant la charge de la supercapa, interrompant l'alimentation. Il est donc indispensable de réguler la tension fournie par la turbine en amont, notamment pour éviter d'endommager le circuit si la tension dépasse les 40 V.

Afin de savoir dans quel état se trouve le chargeur des LED avec différents codes sont présents sur la carte. Une LED STAT et une LED PG, STAT indique l'état de la charge, si elle est éteinte la capacité n'est pas chargée, si elle est allumée la charge est complète. PG indique si l'alimentation est correcte, elle s'allume lorsque c'est le cas et clignote en cas de problème (alimentation trop faible ou trop forte).

Des tests ont été menés pour évaluer le comportement de la turbine et l'efficacité de la recharge de la supercapacité. Le matériel utilisé comprenait une alimentation stabilisée pour alimenter le moteur, un moteur pour entraîner la turbine, un multimètre, ainsi qu'un oscilloscope de type Picoscope (oscilloscope compact connecté à un PC via USB). Une série de tests a été réalisée afin de caractériser et comprendre le comportement de la turbine et du chargeur de supercondensateur.

#### 1. Analyse du signal en sortie de turbine

La turbine a été alimentée par une alimentation de laboratoire réglée à 4 V. À vide, elle génère une tension d'environ 12 V en sortie. Ce premier test visait à observer le comportement de la turbine seule et à confirmer sa capacité à générer une tension suffisante pour alimenter le système en aval.

## 2. Étude du système complet

Les cycles successifs de charge et de décharge du condensateur ont ensuite été observés. Ces cycles sont visibles sur les courbes relevées à l'oscilloscope, notamment en sortie de la turbine avant le chargeur. Cependant, la vitesse de charge s'est révélée insuffisante pour garantir un fonctionnement stable du système.

## 3. Optimisation du courant de charge (paramètre ISET)

Le chargeur permet également de régler la tension de sortie ainsi que le courant de charge à l'aide de deux potentiomètres.

Pour régler la tension de sortie, le chargeur est alimenté par une alimentation stabilisée de 12 V. La tension de sortie est ajustée à l'aide d'un multimètre en mode voltmètre, connecté à la sortie du module. Le premier potentiomètre est réglé à l'aide d'un tournevis jusqu'à obtenir une tension de 3,8 V.

Le réglage du courant de charge se base sur la datasheet du contrôleur de charge BQ25173, selon la formule suivante :

$$I = \frac{K_{set}}{R_{set}} \text{ avec } K_{set} = 300$$

- $R_{SET} = 6 \text{ k}\Omega \Rightarrow I \approx 0.05 \text{ A}$  | Supercondensateur non chargé en 10 minutes.
- $R_{SET} = 4.7 \text{ k}\Omega \Rightarrow I \approx 0.06 \text{ A}$  | Charge légèrement plus rapide, mais encore insuffisante.
- $R_{SET} = 1.5 \text{ k}\Omega \Rightarrow I \approx 0.2 \text{ A}$  | Aucune amélioration significative.
- $R_{SET} = 780 \Omega \Rightarrow I \approx 0.385 \text{ A}$  | Toujours pas de charge satisfaisante.

Ces résultats ont montré que, malgré l'augmentation du courant de charge, la vitesse de chargement de la supercapacité restait trop faible. Une hypothèse a été formulée : augmenter la tension de sortie du chargeur pourrait améliorer la vitesse de charge.

D'après la datasheet de la supercapacité utilisée, celle-ci peut être chargée avec un courant maximal de 0,1 A et une tension de 3,8 V.

Soit pour un courant de charge  $I = 0,1 \text{ A}$  on en déduit :

$$R_{set} = \frac{K_{set}}{I}, \quad R_{set} = \frac{300}{0.1} = 3000$$

Le second potentiomètre est donc ajusté hors tension à l'aide d'un multimètre en mode ohmmètre, jusqu'à obtenir une résistance de 3000 ohms.

Un test complémentaire a donc été effectué avec une sortie réglée à 3,8 V pour un courant de 0.1 A. Résultat : en 10 minutes, la tension du supercondensateur est passée de 1,64 V à seulement

2,145 V, soit un gain de 0,5 V en 10 minutes, ce qui reste insuffisant pour alimenter efficacement le système.

#### 4. Limites de la turbine

Il a également été observé que la fréquence de rotation de la turbine, et donc la tension qu'elle produit, varie en fonction de la consommation de courant par la carte. Cette instabilité ne pose pas énormément de problème au niveau de la tension mais pourra causer par la suite des complications pour la lecture de fréquence.

Pour identifier la source des limitations, l'alimentation a été directement reliée au chargeur de supercapacité. Dans ce scénario, le système a fonctionné correctement, suggérant que la turbine seule ne génère pas un courant suffisant.

Des mesures ont permis de constater que lorsque le courant fourni est inférieur à 0,1 A la tension de sortie du chargeur s'effondre et la charge du supercondensateur stagne à 2,4 V. À titre de comparaison, en alimentation directe, le même chargeur permettait une charge complète.

Avec le moteur utilisé, la tension maximale générée par la turbine atteignait environ 12 V. Toutefois, les premiers résultats se sont avérés peu satisfaisants, le temps de charge de la supercapacité était très long et la tension à ses bornes ne dépassait pas 2,4 V, malgré plusieurs longues minutes de fonctionnement.

Une mesure du courant délivré par la turbine à l'aide d'un ampèremètre a permis d'identifier une intensité insuffisante ce qui explique la durée du temps de charge :

- Sans supercapacité (circuit ouvert) : environ 1,4 mA.
- Avec supercapacité en charge : environ 0,9 mA au moment de la mesure (en augmentation mais ne dépassera pas 1,4 mA.

En courant continu (CC), pour une tension d'alimentation du moteur d'environ 12 V, la turbine délivrait approximativement 20 mA.

Il a été supposé que le comportement instable du système pouvait être lié à un signal trop bruité en sortie de la turbine, perturbant le chargeur. Pour tenter d'y remédier, différents condensateurs de filtrage ont été ajoutés en sortie de turbine afin d'essayer de lisser le signal tout en augmentant l'intensité du signal :

- 100  $\mu$ F + 100 nF : courant mesuré à 8,4 mA, consommation de 0,22 A sur l'alimentation.
- 470  $\mu$ F + 100 nF : courant mesuré à 10 mA, consommation de 0,23 A.

#### Test de charge de la supercapacité

- Condensateur de filtrage : 470  $\mu$ F
- Supercapacité initialement à : 1,8 V
- Courant de charge mesuré : 11 mA

Résultat : la tension de la supercapacité est montée à environ 3 V, mais la charge n'était toujours pas complète au bout de 14 minutes. L'objectif étant d'atteindre une charge complète en moins de 10 minutes l'observation a été arrêtée

Dans une nouvelle tentative, les modifications suivantes ont été apportées :

- Ajout d'une diode en sortie de la turbine pour éviter des retours.
- Condensateur de filtrage de 570  $\mu\text{F}$  placé après la diode.
- Tension initiale de la supercapacité : 0,9 V.
- Courant mesuré sur l'ampèremètre : 25 mA.

Malgré ces améliorations, après 11 minutes, la supercapacité n'était toujours pas complètement chargée. Afin d'augmenter la vitesse de rotation de la turbine, la puissance du moteur a été augmentée.

Cependant, cela a révélé une nouvelle contrainte :

En augmentant la capacité de filtrage, le courant délivré augmente, mais cela augmente également la charge à entraîner, donc le couple résistant. Le moteur chauffe considérablement, ce qui devient problématique pour sa durabilité et la sécurité du système.

Ces observations ont mis en évidence une limite importante quant à la simulation par moteur, pour améliorer la puissance électrique récupérée, il faut soit optimiser la conversion mécanique/électrique (meilleure turbine, meilleur rendement), soit trouver un compromis entre le courant disponible et le couple imposé au moteur.

La turbine peut produire jusqu'à 80V, or le chargeur de super capa n'est protégé que jusqu'à 40V et ne fonctionne que jusqu'à 18V à la suite de quoi il se met en sécurité. Afin de protéger le circuit contre les surtensions, plusieurs tests ont été réalisés avec des diodes TVS (diodes de suppression de tension transitoire), tout en étudiant leur impact sur le comportement de charge. En parallèle, des essais ont été menés pour évaluer l'influence des condensateurs de filtrage sur le couple moteur, le courant de charge, et la vitesse de montée en tension de la supercapacité.

Les diodes TVS sont des composants de protection utilisés pour protéger des surtensions contrairement à un régulateur de tension, qui sert à stabiliser une tension en continu. Cependant cela inclut une certaine perte de puissance, c'est pourquoi il a pour le moment été décidé d'utiliser des TVS qui sont normalement conçues pour intervenir uniquement lors de pics de tension très courts. La diode est inactive en fonctionnement normal, elle se comporte comme un circuit ouvert tant que la tension reste inférieure à un certain seuil. Lorsque ce seuil est dépassé la diode va se fermer reliant peu à peu le circuit à la masse. La coupure de tension n'est pas instantanée et suit une certaine courbe sur laquelle sont compris plusieurs seuils.

- $V_{rwm}$  : la tension à partir de laquelle la diode commence à se fermer pour relier le système à la masse, en dessous de cette tension la diode est inactive.
- $V_{br}$  : la tension de claquage. Lorsque la tension appliquée dépasse ce seuil, la diode commence à conduire fortement.
- $V_{cl}$  : la valeur à laquelle la diode est entièrement passante, c'est la valeur max de la TVS, le circuit ne pourra pas dépasser cette valeur



Dans un premier temps, il a été nécessaire d'identifier les diodes TVS disponibles dans l'entreprise, qui n'étaient référencées que par leur référence RS, sans documentation associée. Afin de faciliter leur réutilisation pour les futurs projets, il m'a également été demandé de recenser leurs principales caractéristiques techniques. Dans le cadre du projet, les diodes recherchées devaient être unidirectionnelles et capables de limiter les surtensions autour de 18 V. Plus d'une dizaine de références étaient disponibles, mais seules les versions unidirectionnelles ont été retenues pour l'étude, les bidirectionnelles ayant été écartées.

Les diodes unidirectionnelles étant celles-ci :

Référence	V <sub>rw</sub>	V <sub>br</sub>	V <sub>cl</sub>
805-0586	20	22.2 – 24.5	32.4
802-2373	70	77.8 – 86.0	113
803-1986	5.8	6.45 – 7.14	10.5
610-4451	10	11.1 – 12.3	17
PTVS12VSIUR115	12	14	19.9
628-7512	7.5	8.33 – 9.21	12.9
628-7657	15	16.7 – 18.5	24.4
628-7304	5	6.4 – 7.07	9.2

Le chargeur se mettant en sécurité à partir de 18V, 3 diodes ont été retenues pour réaliser des tests la 610-4451, la PTVS12VSIUR115 et la 628-7657.

Pour plus de simplicité les diodes sont nommées par leur valeur V<sub>rw</sub> et V<sub>cl</sub> plutôt que leur référence.

#### Test avec diode TVS 12–19,9 V

- Tension d'alimentation : 9 V
- Courant mesuré : 20 mA
- Tension initiale supercapacité : 1,3 V

Dans ce test, un condensateur de 100  $\mu$ F était présent en sortie de turbine. Après 5 min 30 s, la supercapacité atteignait 3 V. Pour évaluer l'impact du condensateur sur le système, celui-ci a été temporairement retiré.

#### Observations :

- Sans le condensateur, le couple résistant du moteur est significativement réduit.
- La tension générée augmente légèrement.
- La différence de courant mesuré est faible : une perte de seulement 4 mA en l'absence du condensateur.

#### Comparaison avec et sans condensateur de 100 $\mu$ F (durée de 10 minutes, TVS 12–19,9 V) :

- Tension de départ : 1,7 V

- Tension d'alimentation : 7 V

Tension après 10 minutes :

- Sans capacité : 2,4V
- Avec capacité : 2,8V

L'ajout du condensateur de filtrage améliore la vitesse de charge, malgré une légère augmentation du couple. Il permet également de mieux stabiliser le signal issu de la turbine, ce qui profite à l'efficacité du chargeur.

Des observations complémentaires ont été faites pour analyser le comportement du système pendant la charge :

- Sans condensateur, la supercapacité tend à se décharger plus rapidement qu'elle ne se recharge.
- Avec condensateur, le système parvient à charger la supercapacité plus efficacement, et ce, sans impact notable sur la turbine. Le couple moteur reste faible, même avec le filtrage, ce qui en fait une solution équilibrée.

Test avec diode TVS 15–24 V

- Tension d'alimentation : 9 V
- Courant mesuré : entre 22 mA et 24 mA

Ce test visait à évaluer une plage de protection plus élevée. Les premières mesures indiquent une légère hausse du courant en comparaison avec la TVS précédente, mais des tests plus poussés seraient nécessaires pour déterminer le compromis optimal entre protection, rendement et stabilité.

Les tests ont montré que l'ajout d'un condensateur de filtrage en sortie de turbine permet d'améliorer la stabilité du signal et la vitesse de charge, sans engendrer un couple excessif. Le choix de la diode TVS influe également légèrement sur le courant délivré, ce qui mérite une analyse comparative plus approfondie.

Des tests ont été réalisés pour vérifier si le système pouvait bien charger complètement la supercapacité, puis fonctionner correctement lors de plusieurs cycles de charge et de décharge.

Objectifs des tests

1. Vérifier si la supercapacité peut être chargée complètement, quel que soit le temps nécessaire.
2. Si c'est le cas, réaliser plusieurs cycles complets de charge/décharge pour mesurer :
  - Le temps de charge complet.
  - Le temps de décharge (via le système de consommation).



3. Évaluer la consommation réelle du système en phase de décharge. Cela inclut les pertes, notamment celles liées à la présence de LEDs parasites ou de composants inutiles.

Limite rencontrée

Le système n'avait pas encore atteint la charge complète lors des premiers essais, et un comportement anormal du moteur (bruits irréguliers, perte de performance, tremblement) a été constaté. Ce problème semblait venir de la structure de support en impression 3D.

Les tests ont alors été mis en pause le temps de réparer le support du moteur à l'aide de lime et de colle forte pour stabiliser la structure avant de reprendre.

Bien que le système, dans son état actuel, ne soit pas encore énergétiquement rentable, cela s'explique principalement par les limitations du moteur utilisé pour simuler la turbine. Ce dernier présente un couple très faible et une vitesse de rotation limitée.

Cependant, dans un contexte réel, la force hydraulique (pression et débit d'eau dans une canalisation) exercera un couple bien supérieur à celui généré par le moteur d'essai. Cette force devrait aisément compenser les pertes induites par la force contre-électromotrice de la turbine, assurant ainsi un fonctionnement beaucoup plus efficace.

Des tests ont également été effectués avec un moteur plus puissant, offrant un meilleur couple et une vitesse légèrement plus élevée. La turbine délivre une tension suffisante pour charger la supercapacité sans mettre le chargeur en sécurité quelle que soit la vitesse du moteur grâce à la TVS.

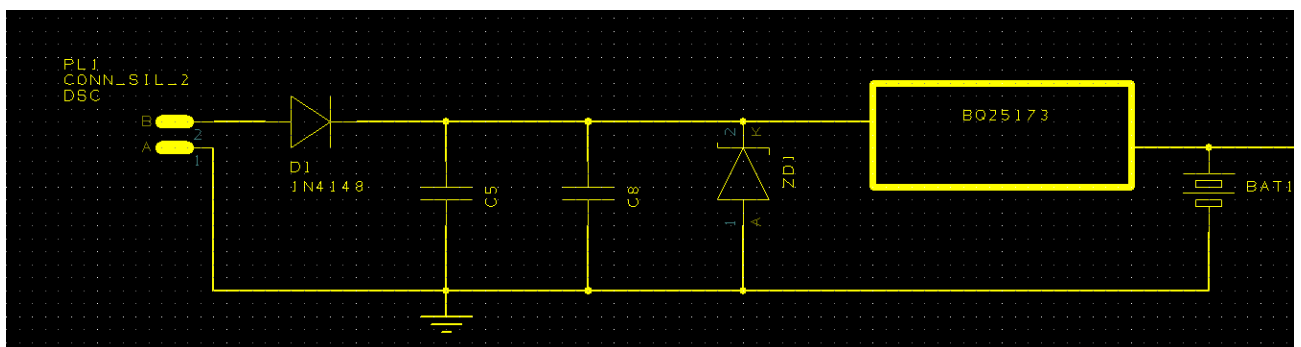


Figure 4 Schéma de câblage du chargeur

#### Etude de la mesure de débit

Afin de mieux comprendre le fonctionnement de la turbine utilisée pour mesurer le débit, une première étape a consisté à démonter celle-ci pour en observer l'intérieur. La turbine est constituée de 9 bobines réparties en 3 groupes de 3, chaque groupe étant connecté à une petite

carte. Un seul des trois groupes est utilisé pour la mesure : un fil y est soudé et connecté à l'électronique, permettant de récupérer un signal de fréquence en sortie.

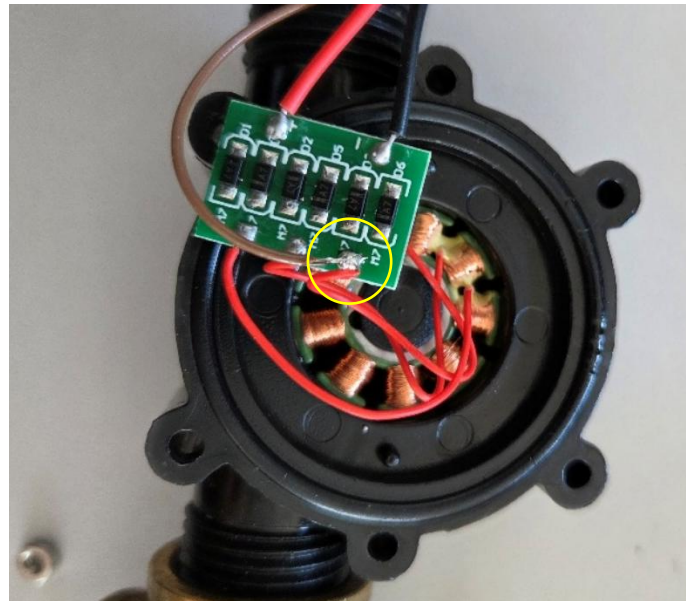


Figure 5 Intérieur de turbine

Un premier circuit avait déjà été réalisé pour effectuer cette mesure de fréquence, mais il présentait plusieurs défauts techniques (voir annexe 2).

Les premiers tests ont été réalisés avec ce montage afin de vérifier son bon fonctionnement et obtenir des résultats sur lesquels se baser. Ces résultats ont montré un comportement relativement satisfaisant, le système parvient à charger le supercondensateur même avec le robinet complètement ouvert, et ce en seulement quelques minutes. Ces tests ont également confirmé que la puissance générée par l'eau était bien suffisante pour alimenter le circuit et dissiper les craintes rencontrées avec les moteurs quant à la réaction contre-électromotrice.

Les mesures comme le temps de charge et fréquence du signal ont été réalisées par la suite, dans des conditions plus contrôlées.

Afin de vérifier la véracité des mesures un vrai débitmètre est intégré au système d'eau.

Les essais ont été réalisés dans un environnement de test improvisé. Un montage avec bassines, tuyaux, une pompe à eau et un vrai débitmètre a été mis en place pour simuler l'écoulement de l'eau à travers la turbine. Ces premiers tests avaient pour but de valider le bon fonctionnement général du système : vérifier que la carte électronique ne se met pas en sécurité, que la charge fonctionne, et qu'aucune fuite ou comportement anormal n'apparaît.

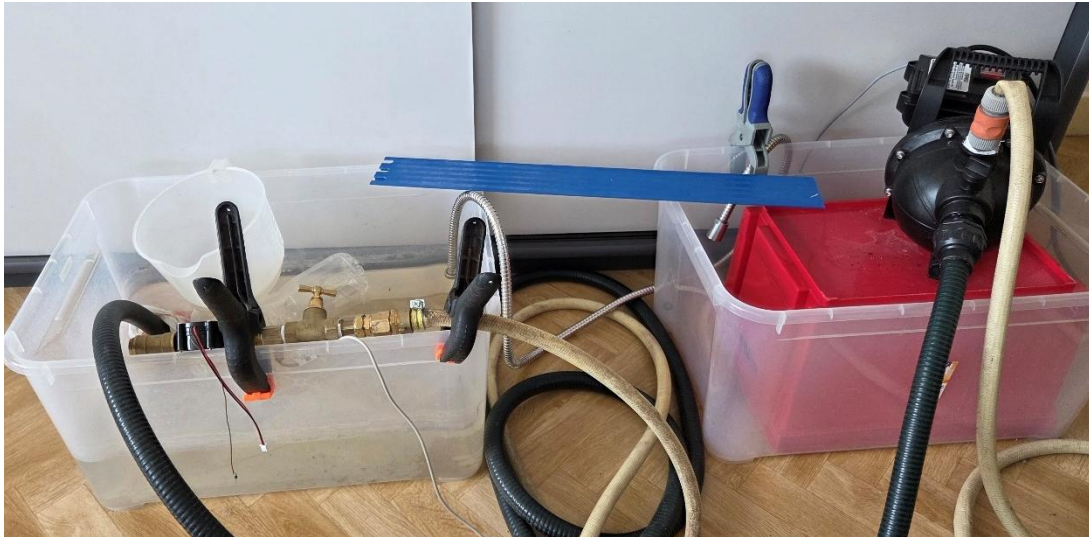


Figure 6 Installation pour simulation en condition presque réelle

La fréquence mesurée à la sortie du débitmètre de référence permet d'en déduire le débit en litres par minute (L/min). Selon les spécifications indiquées sur l'étiquette du débitmètre, celui-ci génère 495 impulsions par litre. La relation entre la fréquence  $f$  (exprimée en Hz) et le débit s'exprime donc par la formule :

$$\text{Débit (L/min)} = f * \frac{60}{495}$$

Dès les premiers essais un constat a été fait, une zone de perturbation sur la fréquence qui correspond à la tension à laquelle la TVS commence à s'activer. Cette zone de perturbation pouvait provoquer de nombreux problèmes notamment pour la lecture du débit puisqu'utilisant la fréquence. Un long travail d'étude et de recherche a alors commencé. Cet artefact est présent sur les tests qu'ils soient réalisés à vide, avec le supercondensateur en charge ou chargé. (Voir annexe 3)

Cet artefact autour des 55 Hz était présent sur la quasi-totalité des tests, jusqu'à ce qu'une solution soit trouvée, peu importe le montage cette zone d'interférence était toujours présente, ce malgré les nombreux tests réalisés.

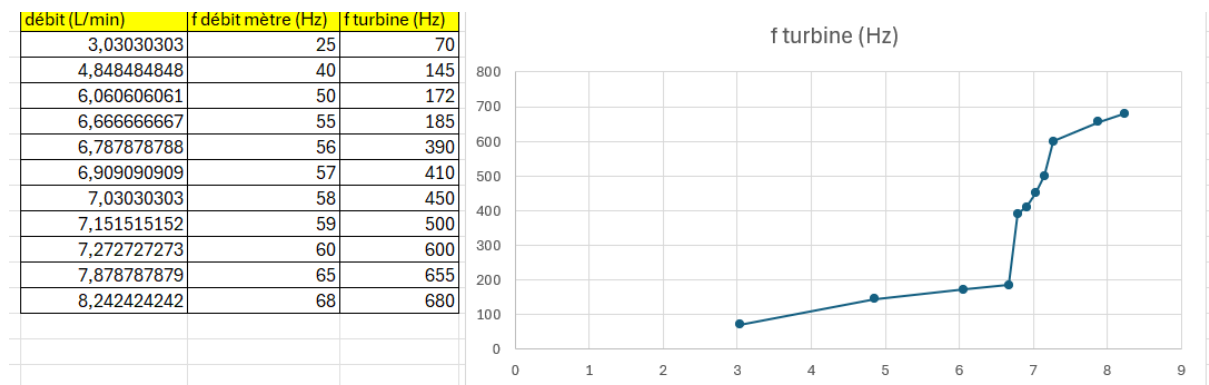


Figure 7 débit en fonction de la fréquence

Un premier test a été réalisé en retirant le transistor et la résistance de 100 kΩ placée en sortie du MOSFET. Cette configuration avait pour objectif de simplifier le montage afin d'observer le signal brut en sortie des bobines de la turbine. Le pont diviseur de tension, composé de deux résistances de 500 kΩ, a été conservé pour abaisser la tension pouvant atteindre jusqu'à 80 V. Une capacité de 100 nF a également été placée en sortie du pont afin de filtrer les parasites et lisser le signal. Bien que les courbes obtenues soient relativement linéaires, le signal observé à l'oscilloscope n'était pas suffisamment stable pour être exploitable de manière fiable. (Voir annexe 4)

Un second test a été mené en conservant cette configuration RC, mais en remplaçant le tuyau initial par un modèle de diamètre plus large. Cette modification visait à augmenter le débit d'eau et à réduire les perturbations éventuelles du flux, possiblement induites par les caractéristiques du tuyau précédemment utilisé. Toutefois, les résultats sont restés similaires au test précédent, sans amélioration significative de la stabilité du signal. (Voir annexe 5)

Une nouvelle version du système électronique a donc été conçue, cette fois en réintégrant un transistor MOSFET de type N-channel, tout en conservant les éléments efficaces du montage précédent. Après le pont diviseur de tension (500 kΩ + 500 kΩ) et la capacité de 100 nF, le signal est injecté dans la grille du MOSFET. En sortie, sur la broche Drain, une résistance de 10 kΩ est reliée au supercondensateur, servant de tension de référence pour la mesure de fréquence. Une diode Zener est également ajoutée en sécurité afin de limiter les surtensions. Cette nouvelle configuration a permis d'obtenir un signal plus propre, exploitable et stable à l'oscilloscope, avec des courbes de mesure linéaires.

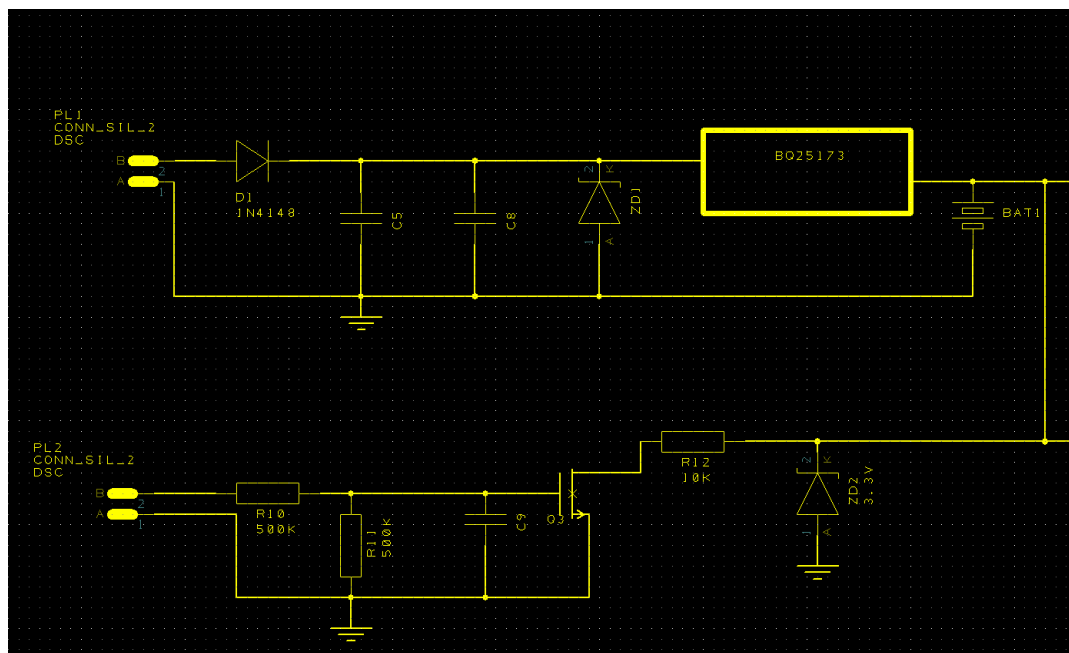


Figure 8 Schéma de câblage du système finale

Au total, quatre courbes ont été tracées afin d'analyser le comportement du système dans différentes conditions. Pour chacune des deux supercapacités testées, deux configurations ont été comparées : une première avec la supercapacité en cours de charge, et une seconde avec la supercapacité déjà pleine. Ces mesures ont permis de valider la cohérence du fonctionnement du système de mesure, quel que soit l'état de charge initial de la supercapacité. (Voir annexe 6)

### *Etude de l'afficheur*

En l'absence temporaire du microcontrôleur NXP initialement prévu pour le projet, une carte NodeMCU V3 (ESP8266 non officielle) a été utilisée à titre de remplacement. Cette solution, uniquement temporaire, a permis de poursuivre le développement en attendant la réception du matériel définitif. Elle a également offert l'opportunité de vérifier le bon fonctionnement des différents composants dans un environnement de développement maîtrisé, tout en facilitant l'élaboration d'un code simple, destiné à être adapté par la suite.

L'écran e-paper fourni ne possédait aucune référence visible ni datasheet disponible, ce qui a considérablement compliqué son intégration. Il a donc fallu procéder en testant différentes bibliothèques Arduino compatibles avec les écrans e-paper, tester toutes les bibliothèques et exemples possibles et les tester un par un. Plusieurs obstacles sont apparus :

- Certaines bibliothèques étaient obsolètes ou incompatibles avec la version actuelle de l'IDE Arduino ou de l'ESP8266.
- D'autres entraient en conflit avec les pilotes existants.
- Les exemples de code fournis ne fonctionnaient pas directement, ce qui a nécessité des adaptations manuelles.

Grâce à l'aide de Bernard, une documentation technique de l'écran réalisée par l'un de ses contacts ayant déjà travaillé précisément sur le même écran e-paper avec un NodeMCU v3 a pu être obtenue. Cette documentation a permis de confirmer le schéma de câblage exact et d'identifier la bibliothèque logicielle la plus adaptée. (Voir annexe 7)

Par la suite, plusieurs ajustements ont été nécessaires : désinstallation de certaines versions de drivers, modifications de configurations, et résolution de conflits d'incompatibilité. Ces problèmes ont été progressivement résolus grâce aux ressources trouvées sur des forums spécialisés tels que GitHub, Arduino ou Stack Overflow.

Une fois ces étapes finalisées, le code source a pu être adapté, permettant ainsi l'affichage des premières données sur l'écran.

Une fois l'écran opérationnel, les premières fonctionnalités du programme ont été progressivement intégrées. Celui-ci assure à la fois la mesure du débit d'eau, le suivi de la consommation, la lecture de la tension de la supercapacité et l'affichage de l'ensemble de ces données, à la fois sur un moniteur série et sur un écran e-paper.

## 1. Mesure du débit

La turbine installée dans le système génère une série d'impulsions électriques, proportionnelles au débit d'eau circulant. Ces impulsions sont détectées via un front montant sur une entrée numérique de la carte (D6 de la NodeMCU V3). Le programme comptabilise le nombre d'impulsions par seconde, ce qui permet de calculer la fréquence de rotation de la turbine, exprimée en hertz (Hz). Un coefficient de conversion, défini en fonction des mesures réalisées précédemment sur la turbine, permet ensuite de convertir cette fréquence en débit instantané, exprimé en litres par minute (L/min). Le coefficient de conversion est une moyenne des équations obtenues sur Excel puis adapté.

```
float convertirEnDebit(float freq) {  
    float d1 = (freq + 31.15) / 27.288;  
    float d2 = (freq + 22.724) / 25.76;  
    float d3 = (freq + 30.456) / 27.798;  
    float d4 = (freq + 15.099) / 24.852;  
    return (d1 + d2 + d3 + d4) / 4.0;  
}
```

Figure 9 Equation de conversion en débit

## 2. Estimation de la consommation d'eau

Afin d'estimer la consommation totale d'eau, le programme effectue une moyenne des débits mesurés sur les 60 dernières secondes. Ce débit moyen est ensuite intégré dans le temps pour obtenir un volume cumulé, exprimé en litres. Cette valeur est mise à jour en continu, permettant de suivre en temps réel la quantité d'eau écoulée depuis le début du lancement du programme.

## 3. Lecture de la tension de la supercapacité

La supercapacité utilisée dans le système peut atteindre une tension maximale de 3,85 V, cependant l'entrée analogique A0 utilisée de la carte NodeMCU ne peut supporter que des tensions comprises entre 0 et 1 V. Pour permettre une mesure sécurisée, un pont diviseur de tension a été mis en place à l'aide de deux résistances de 500 kΩ. Ce dispositif réduit la tension d'entrée par un facteur d'environ 5. Une capacité de filtrage de 100 nF est également installée en sortie du pont afin de stabiliser la mesure, en limitant le bruit du signal.

La tension analogique lue (comprise entre 0 et 1024 en valeur brute) est convertie en tension réelle à l'aide du facteur de division mesuré expérimentalement. Cette tension est ensuite exprimée en pourcentage de charge de la supercapacité, sur une base maximale de 3,85 V.



#### 4. Affichage des données

Les différentes informations collectées sont affichées sur deux supports distincts :

- Sur le moniteur série (pendant la phase de développement) :
  - Fréquence de rotation de la turbine (Hz)
  - Débit instantané (L/min)
  - Volume d'eau consommé (L)
  - Tension réelle de la supercapacité (V)
  - Niveau de charge (%)
- Sur l'écran e-paper :
  - Débit (L/min)
  - Fréquence (Hz)
  - Volume total consommé (L)
  - Pourcentage de charge de la supercapacité (%)

```
=====
Pulses : 496 | Frequence : 248.00 Hz | Debit : 10.33 L/min
Tension lue sur A0 : 0.606 V | Tension capa : 3.091 V | super capa chargé à : 81.34 %
=====
Pulses : 494 | Frequence : 247.00 Hz | Debit : 10.29 L/min
Tension lue sur A0 : 0.606 V | Tension capa : 3.091 V | super capa chargé à : 81.34 %
=====
Pulses : 496 | Frequence : 248.00 Hz | Debit : 10.33 L/min
Tension lue sur A0 : 0.606 V | Tension capa : 3.091 V | super capa chargé à : 81.34 %
=====
```

Figure 10 Affichage du terminale série

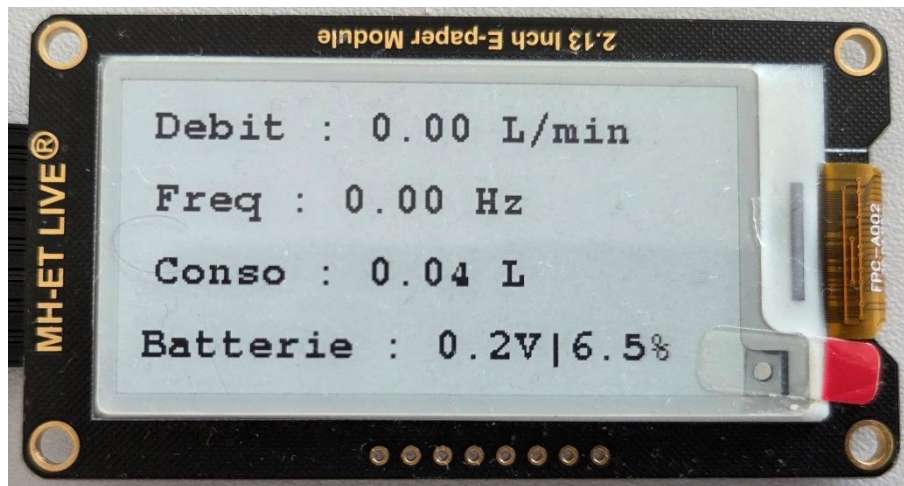


Figure 11 affichage écran e-paper

## 5. Incident matériel

Lors des essais de l'écran e-paper, un incident matériel est survenu, la diode TVS a grillé. Ce type de défaillance peut être révélateur de plusieurs problèmes potentiels. En observant la datasheet des diodes TVS un autre facteur était à prendre en compte, la puissance maximale dissipée, un dépassement prolongé de cette valeur peut entraîner l'endommagement du composant.

Par ailleurs, les diodes TVS sont conçues pour absorber des surtensions instantanées, par exemple si la foudre frappe une maison, la diode TVS va rediriger la tension vers la masse. Or, dans le cadre du projet, la diode était sollicitée de manière quasi continue, ce qui ne correspond pas à son usage prévu, et pourrait expliquer son endommagement.

Afin de continuer les tests la diode a été changée et un modèle plus gros a été choisi, la 628-7657, diode TVS 15-24V. Ce changement a impliqué de refaire les mesures et tests précédents, les caractéristiques électriques pouvant légèrement varier d'un modèle à l'autre. Les essais ont ensuite pu reprendre avec la nouvelle configuration.

## Missions réalisés extérieur au projet

Durant le stage la mission a parfois été mise en pause afin de soulager l'équipe sur différentes tâches dans l'atelier, d'aider à préparer du matériel pour aider l'équipe à gagner du temps sur certaines commandes.



## 1. Soudure de carte électronique

Plusieurs sessions de soudure ont été effectuées, portant sur différents types de composants électroniques, parmi lesquels :

- Des composants discrets classiques (LEDs, résistances, potentiomètres, etc.),
- Des ponts et connecteurs standards,
- Des sous-ensembles électroniques complets destinés à des commandes en cours.

Ces interventions ont permis d'alléger la charge de travail de l'équipe et de contribuer au respect des délais de livraison.

## 2. Fabrication de câbles d'antenne

Une participation active a également été apportée à la réalisation de câbles d'antenne, une opération nécessitant précision, rigueur et maîtrise des procédés de câblage. En tout, six câbles ont été confectionnés :

- Quatre câbles d'une longueur de 2 mètres,
- Deux câbles de 10 mètres.

La fabrication d'un câble d'antenne suit un processus rigoureux, organisé en plusieurs étapes clés.

Tout d'abord, le câble est découpé à la longueur souhaitée à partir d'une bobine. Les extrémités sont ensuite préparées en retirant la gaine extérieure, ce qui permet d'accéder aux conducteurs internes, composés de cuivre, de fils d'aluminium et d'isolants.

La seconde phase concerne le traitement des extrémités du câble. Du côté des connecteurs, les fils sont découpés, dénudés, puis les connecteurs métalliques de type « pins » sont sertis sur les conducteurs. Du côté de l'antenne, les fils nus sont tressés pour faciliter la manipulation. Une découpe partielle de la gaine permet d'exposer les conducteurs du fil noir sur lequel est ensuite soudé le fil blanc et le fil de masse. Une diode de protection est également insérée sur le fil vert, avec la cathode orientée vers le câble et l'anode vers l'antenne.

Enfin, les finitions consistent à effectuer le sertissage final des connecteurs, à insérer les conducteurs dans un connecteur plastique, puis à poser des gaines thermiques sur les zones sensibles. Les câbles sont ensuite testés avec des tests de continuité puis testés sur des cartes.

Même si cette opération est relativement simple et manuelle, elle demande tout de même de l'attention. Une erreur de câblage, une soudure mal faite ou un sertissage incorrect peut suffire à empêcher le bon fonctionnement du système radio, en provoquant par exemple des pertes de signal ou une absence totale de communication.

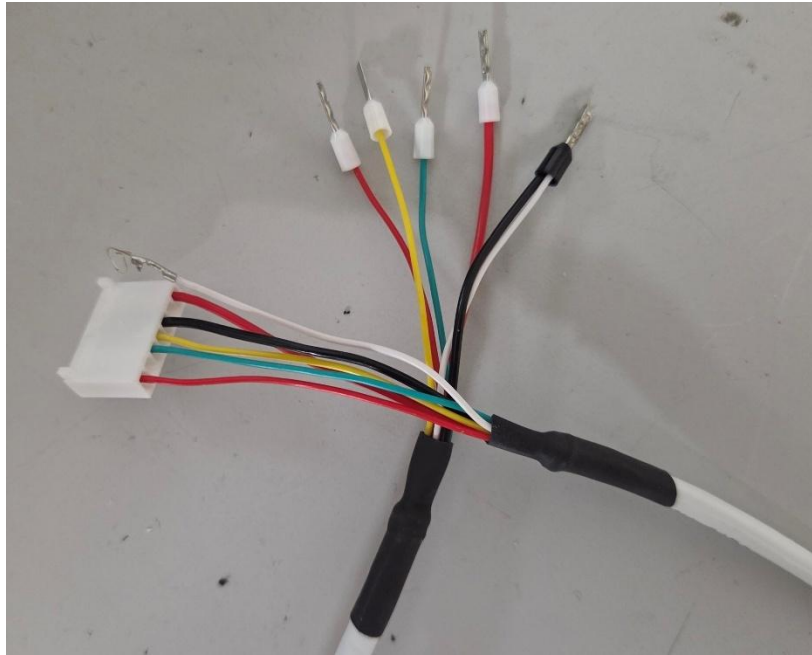


Figure 12 Câble d'antenne

### *Recherche des composants*

#### 1. Sélection et approvisionnement des composants

Dans le cadre du stage, il a été nécessaire de commander différents composants, à la fois pour pouvoir progresser dans l'avancement du projet et pour optimiser le système existant. Le choix des composants s'était basé sur le stock disponible dans l'entreprise, impliquant une certaine limite de choix et de possibilités. Il a donc fallu, d'une part, rechercher et sélectionner un microcontrôleur adapté aux besoins du projet, en tenant compte des contraintes imposées (comme l'utilisation d'un modèle NXP). D'autre part, compléter les essais par l'achat de composants de test supplémentaires afin d'optimiser le système. Cela inclut notamment des diodes TVS de différentes caractéristiques ou encore d'un autre modèle de supercondensateurs, celui utilisé actuellement présentant plusieurs défauts. Aucun cahier des charges strict n'ayant été fourni au départ, la sélection s'est faite en autonomie en s'appuyant sur les critères techniques validés avec le tuteur de stage.

#### 2. Recherche du microcontrôleur

Le premier code a été réalisé sur une carte NodeMCU V3 à base d'ESP8266. Bien que très accessible ce n'est pas une carte viable pour un produit définitif, pour plusieurs raisons :

- Consommation trop élevée, incompatible avec un système autonome

- Nombre de broches insuffisant pour connecter l'ensemble des périphériques (écran, capteur, radio, sonde thermique...).
- Il s'agit d'une carte de développement complète, encombrante et non exploitable dans un projet comme celui-ci. Le but étant d'intégrer uniquement un microcontrôleur sur un PCB personnalisé, dans un souci de compacité.

Le choix du microcontrôleur était l'une des étapes la plus importante du projet. La seule contrainte imposée était de sélectionner un modèle provenant du catalogue de NXP Semiconductors, ce qui limitait les options mais laissait un choix très large vu la quantité de références disponibles. Il a donc fallu commencer par trier en fonction des besoins du projet : tension d'alimentation, nombre de broches d'entrée/sortie (numériques et analogiques), présence d'un mode veille et consommation en veille, présence de bus SPI, etc. N'ayant que peu de connaissance sur le sujet, afin de savoir dans quelle direction commencer à chercher une grande partie des recherches s'est réalisé sur des forums. Un premier microcontrôleur Kinetis a été repéré, mais après discussion avec le tuteur, il s'est avéré que ce modèle faisait partie d'une ancienne gamme arrêtée. La recherche a donc été recentrée sur un modèle plus récent et toujours supporté, un microcontrôleur MCX C14x/24x/44x.

### 3. Recherche des autres composants

En complément du microcontrôleur, d'autres composants essentiels ont dû être identifiés et commandés afin de poursuivre les tests et améliorer le système en place. Jusqu'alors, le montage reposait principalement sur les composants disponibles dans le stock de l'entreprise.

Les recherches ont porté notamment sur des supercondensateurs, des diodes de protection TVS, des régulateurs de tension ou encore des modules radio. Le régulateur de tension doit permettre de protéger le microcontrôleur, qui est l'élément le plus important et le plus onéreux du système, sans perte excessive. Concernant le supercondensateur, celui utilisé initialement présentait une marge d'erreur de 40 % sur sa capacité, une marge d'erreur bien trop importante pour garantir des mesures fiables. Il a donc été nécessaire de sélectionner un modèle plus fiable.

Un autre critère déterminant était la disponibilité à long terme. Tous les composants choisis devaient être encore en production et accessibles facilement. Cela permet de s'assurer que le produit pourra fonctionner sur le long terme, tout en restant réparable en cas de panne, sans avoir à modifier l'ensemble du circuit ou revoir toute l'architecture.

### 4. Plateformes de recherche et d'approvisionnement

La majorité des composants a été identifiée et sélectionnée via les plateformes suivantes :

- Mouser Electronics, choisi comme fournisseur principal pour centraliser les commandes,
- Farnell et RS Components, pour certaines références spécifiques,
- AliExpress, dans le cas de modules plus accessibles de type grand public

Il a été décidé de regrouper les commandes principalement sur Mouser de façon à limiter les frais de port et à réduire les délais de livraison.

La liste exacte des composants est disponible en annexe. Les types de composants commandés sont les suivantes :

- Supercondensateurs
- Chargeurs pour supercondensateurs
- Diodes de protection TVS
- Régulateurs de tension (buck-boost)
- Modules radio
- Microcontrôleur NXP

Le choix s'est porté sur un modèle de la famille MCX C14x/24x/44x de NXP, en raison de plusieurs atouts adaptés aux contraintes spécifiques du projet. Ce modèle, basé sur un cœur Arm® Cortex®-M0+, présente l'avantage d'avoir une très faible consommation, la présence d'un mode veille, ainsi que deux interfaces SPI indépendantes. Afin de pouvoir réaliser des tests ainsi que l'adaptation du programme Arduino une carte de dev FRDM MCXC242 a été commandée.

### *Etude du microcontrôleur NXP*

#### 1. Changement d'IDE

Le développement sur les microcontrôleurs NXP s'effectue via l'IDE MCUXpresso, un environnement développé par NXP qui offre comme l'IDE Arduino des bibliothèques (en C / C++) et des exemples de code.

La transition entre l'environnement de développement de Arduino à MCUXpresso s'est révélée nettement plus complexe que prévu. Contrairement à Arduino, qui propose une interface épurée et simpliste avec des bibliothèques préconfigurées, MCUXpresso s'appuie sur une approche bien plus technique, avec une interface très chargée et de nombreux paramètres à configurer manuellement.

Parmi les différences notables, la gestion des broches illustre bien ce changement de logique. Arduino permet une configuration simple via de simples fonctions ou des définitions déjà faites dans les librairies tandis que MCUXpresso demande une configuration entièrement manuelle.

Cette différence a eu comme conséquence de devoir étudier rigoureusement chaque composant utilisé afin d'en comprendre le fonctionnement dans les moindres détails. Tache qui n'a pas été de tout repos puisque des composants comme l'écran e-paper ont été commandé sur des sites chinois pas chers. Le composant n'étant donc que peu documenté, aucune datasheet

officielle du produit ou des fiches écrites en chinois. L'absence de document clair a grandement compliqué son intégration notamment au niveau de la communication SPI.

Contrairement à Arduino, qui est un environnement très exploité où des librairies existent pour presque tout cas possible, ici, tout doit être fait à la main. Le développement sur MCUXpresso impose de comprendre en détail comment fonctionne chaque composant utilisé.

Il ne s'agit pas simplement de déclarer une broche ou d'envoyer une commande SPI : il faut étudier les datasheets pour comprendre comment chaque bit est interprété, quelles sont les plages de tension tolérées, quelles séquences d'initialisation sont attendues, et quels timings doivent être respectés.

Pour faire fonctionner l'écran ou tout autre module SPI, il est essentiel de connaître précisément le rôle des différents signaux, l'ordre des octets à transmettre, et la manière dont le périphérique s'attend à recevoir les données. Une simple erreur dans la configuration des horloges ou un léger décalage dans les timings peut suffire à rendre le système instable.

## 2. Adaptation du code

Le code Arduino initial ne pouvait pas être utilisé tel quel dans l'IDE de NXP en raison de la différence de librairies. Les librairies Arduino utilisant elles-mêmes d'autres librairies créant ainsi des dépendances internes rendaient certaines fonctions difficilement transférables vers un environnement plus bas niveau comme MCUXpresso.

La première étape a donc consisté à analyser en détail les bibliothèques utilisées dans le projet Arduino, afin d'identifier les fonctions réellement nécessaires notamment la gestion de l'écran e-paper, la lecture des capteurs, la communication SPI, etc. Une fois les fonctions utiles isolées, certaines commandes propres à Arduino, comme `millis()`, `delay()`, `digitalRead()` ou encore `analogRead()`, ont dû être réécrites à la main en utilisant les registres du microcontrôleur NXP.

Le travail d'adaptation a également impliqué de remonter dans la structure des dépendances pour récupérer uniquement les morceaux de code utiles dans le cadre du projet, sans tout importer afin de garder un code simple et facile à comprendre pour ceux qui passeraient après. L'objectif était de reconstruire une base propre, fonctionnelle, et surtout compatible avec l'environnement MCUXpresso, tout en gardant un comportement aussi proche que possible du code réalisé sur Arduino.

## 3. Problèmes rencontrés lors de l'adaptation

Le passage à un environnement bas niveau s'est révélé bien plus complexe que prévu qui aura pris plus de temps que prévu. Plusieurs difficultés ont freiné l'avancement du développement :

- La prise en main de l'environnement de développement NXP a demandé un temps d'adaptation conséquent. L'IDE, plus complexe, nécessite une bonne compréhension des outils et de l'architecture des projets.
- La configuration des broches du microcontrôleur se fait entièrement à la main, en définissant bit par bit les états nécessaires (repos, front, MSB, LSB, etc.), ce qui rallonge considérablement le temps de mise en œuvre.
- Les ressources et aides disponibles en ligne sont bien moindres comparées à celles d'Arduino qui possède une communauté et un nombre d'utilisateurs largement supérieur ce qui rend le dépannage plus difficile.

Le principal blocage rencontré durant la phase de développement concernait l'intégration de l'écran e-paper dans l'environnement NXP. Malgré un code en apparence correct, une alimentation stable, et un brochage conforme à la documentation, aucun signal SPI n'était visible à l'oscilloscope et l'écran ne se mettait pas à jour.

Le débogage s'est effectué étape par étape, avec l'ajout de points d'arrêt dans le code pour suivre l'exécution ligne par ligne, réaliser des mesures au picoscope sur les pins et pouvoir observer à quel moment le code ne fonctionnait pas correctement. L'objectif était de vérifier l'état de chaque broche au fur et à mesure : alimentation, signaux SPI (SCLK, MOSI), broche de sélection (CS), ligne DC, RESET et BUSY.

C'est ce pin BUSY qui était à la source de tout le problème, le pin BUSY est le seul pin en input et qui ne dépend donc pas du code mais des retours de l'écran suite à ce que le code lui envoie. D'après le fonctionnement prévu, il doit passer à l'état haut (1) lorsque l'écran est disponible, et rester à l'état bas (0) lorsqu'il est occupé. Sur Arduino ce comportement était observé normalement, cependant sur la carte NXP, le pin restait bloqué à 0, empêchant l'envoi de toute commande et bloquant ainsi le programme dans une boucle while dans laquelle le code attend le passage de BUSY à 1.

La tentative de résolution du problème a été progressive. Dans un premier temps, le code a été entièrement retravaillé avec un tri des fonctions réellement utilisées dans les bibliothèques Arduino, et la création d'une librairie personnalisée. Pour réaliser cette étape un schéma a été réalisé afin de mieux comprendre le chemin d'appel des différentes librairies sur Arduino. Les librairies bas niveau comme Arduino.h ou SPI.h n'ont pas été importées. Le chemin des dépendances des lib de l'écran est tel qu'en figure 12. Malgré ce nettoyage et une structure de code plus claire, le problème persistait.

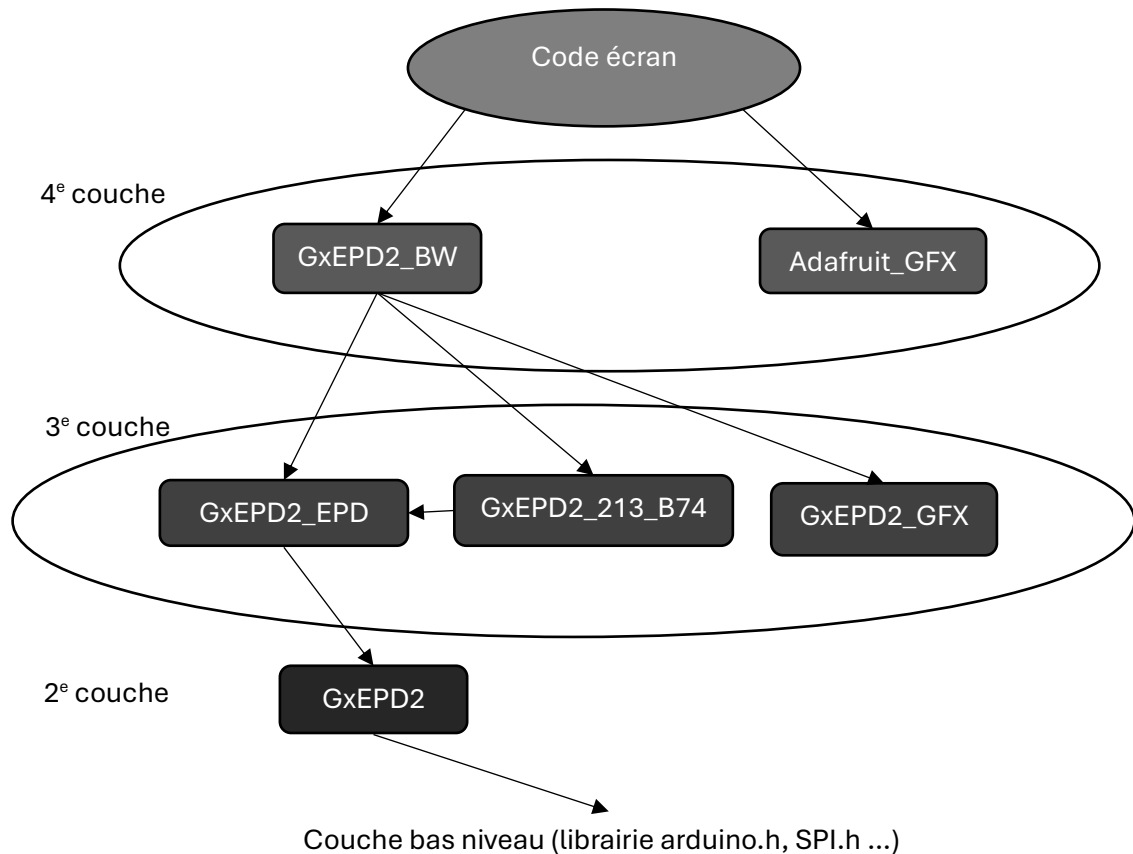


Figure 13 Schéma des dépendances du code de l'écran

Une seconde piste était la configuration matérielle. Certaines broches pouvaient être mal définies ou mal initialisées sur le logiciel (annexe 8). Une fois ces erreurs corrigées, un nouveau schéma de câblage a été réalisé pour avoir une vue claire de l'ensemble du montage (figure 13). Ce schéma de câblage me servira afin de ne pas me perdre dans mon avancement et pouvoir effectuer des modifications plus aisément, mais il servira également à l'entreprise si quelqu'un reprend le travail effectué sur le projet.

Bien que cette approche ait permis de corriger de nombreuses erreurs dans la structure du code et la configuration matérielle, cela ne débloquent toujours pas la situation. Après de multiples tests, il est apparu que toutes les fonctions SPI (reset, envoi de données, initialisation) étaient bien exécutées, sauf une à un endroit spécifique, la ligne `SendCommand()` appelée juste avant la fonction `WaitUntilIdle()`. Cependant la fonction `SendCommand()` était déjà utilisée dans les lignes avant et fonctionnait correctement puisque des changements étaient observés sur les différents pins, de plus le bon envoi du signal est bien observé à l'oscilloscope.



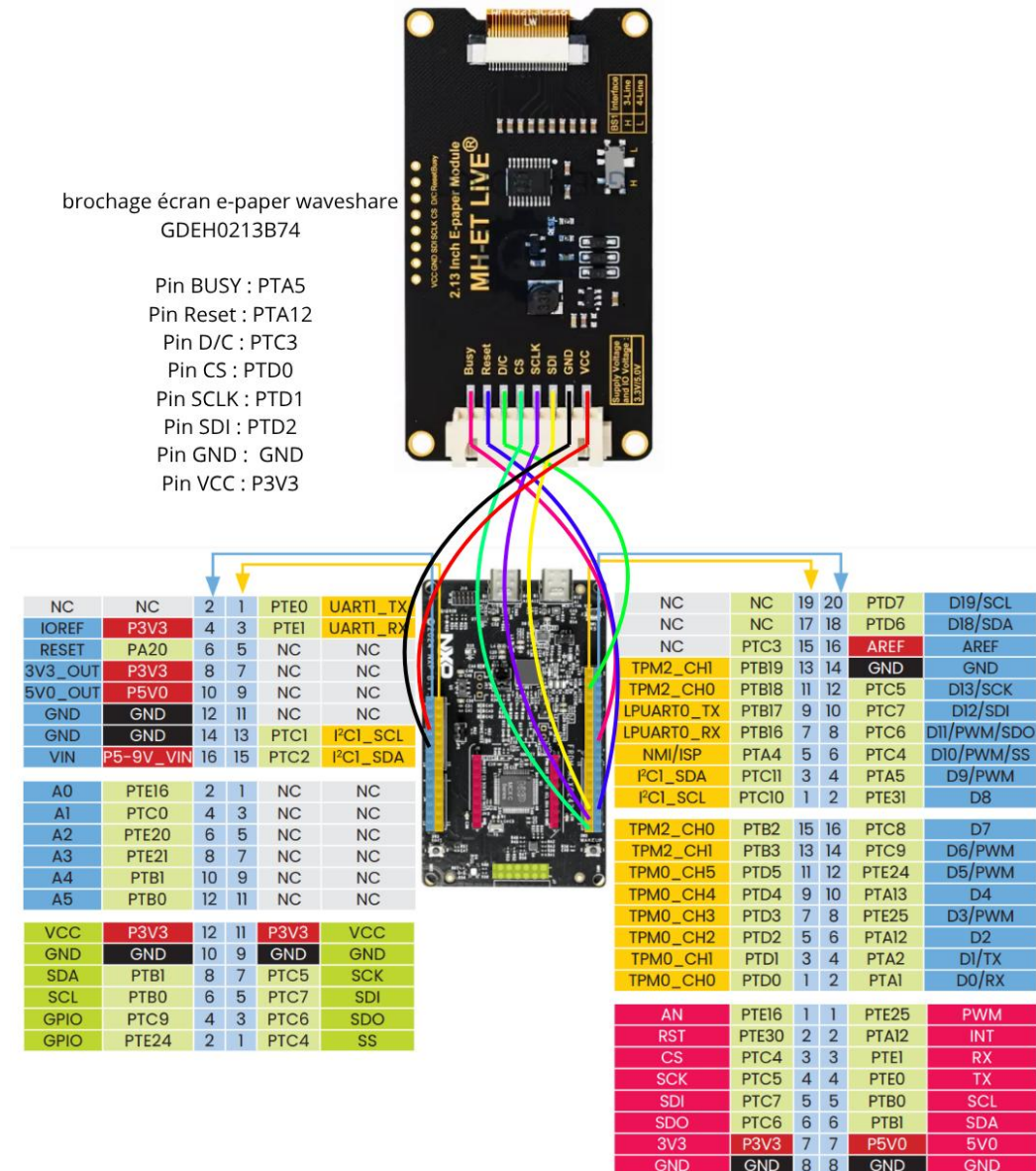


Figure 14 Schéma de câblage NXP

Tout le reste du système semblait fonctionner correctement : câblage validé, signaux mesurés, séquences SPI cohérentes, broches bien définies, alimentation stable. La seule chose qui bloquait était cette ligne de code isolée, sans raison apparente.

Le développement s'est arrêté à cette étape, faute de nouvelles idées de pistes à explorer. De plus, en l'absence de disponibilité du tuteur pour se plonger dans un code aussi spécifique, il n'a pas été possible d'aller plus loin. L'origine exacte du blocage reste inconnue à ce jour, mais l'ensemble du travail réalisé aura permis d'identifier et corriger de nombreuses erreurs et de poser les bases pour la suite du développement.



## Conception du PCB

Après une longue phase de prototypage réalisée sur des plaques de test avec des câbles volants, la conception d'un PCB est une étape importante dans le projet, étape à laquelle le projet commence à se concrétiser, et commence à donner une idée de ce à quoi va ressembler le produit final.

L'idée étant de regrouper tous les composants sur une seule carte en optimisant à la fois la taille, l'alimentation et le routage. Le but est d'avoir une carte compacte, fonctionnelle, propre, et fiable sur le long terme. Ce passage au PCB permet aussi de résoudre un certain nombre de problèmes qu'on peut avoir sur plaque de test, comme les faux contacts, les parasites ou les longueurs de fils trop importantes.

Le logiciel utilisé pour la conception est DesignSpark PCB, assez similaire à KiCad. Il permet de faire le schéma, placer les composants, faire le routage, et générer les fichiers Gerber nécessaires pour faire fabriquer la carte.

Cependant, certains composants clés comme la diode TVS ou le chargeur de supercondensateur BQ25173 n'existaient pas dans la librairie du logiciel. Il a donc fallu les créer à la main. Ce qui implique d'analyser la datasheet afin d'en récupérer les dimensions, les broches, les espacements, etc. Une erreur dans le brochage ou la taille des pads peut rendre la carte inutilisable.

Le cas du BQ25173 est particulier, ce n'est pas un simple composant à intégrer, mais une carte déjà montée, un système complet, avec plusieurs composants dessus dont seules certaines sont réellement utiles dans l'utilisation qui en est fait ici. Il a donc fallu analyser le schéma de câblage de la carte pour repérer les éléments nécessaires puis créer une empreinte avec les bonnes dimensions.

Une fois le PCB finalisé, il ne restera plus qu'à le commander, le souder, et tester tout le système dans des conditions proches de la réalité. Cela permettra notamment de savoir de quelle taille sera le boîtier contenant la carte qui sera également modélisé par MicroBE en plus de pouvoir réaliser des tests ainsi que de potentiels ajustements.

## Bilan

### Présentation des résultats

À mon arrivée dans l'entreprise, le projet n'était que très peu développé, des tests réalisés principalement autour de la turbine et de la possibilité de mesurer le débit d'eau. Aucun réel développement complet et poussé n'avait encore été entrepris dû à un manque de ressources humaines et de l'envergure du projet. Les composants prévus pour l'élaboration du système étaient, pour la plupart, encore à l'état de test ou n'avaient jamais été intégrés. L'objectif de mon stage était donc double : faire avancer concrètement le projet et poser des bases techniques solides, documentées et réutilisables.

Durant la période de stage, plusieurs apports concrets ont été réalisés, à la fois sur le plan technique, structurel et organisationnel.

## 1. Avancées techniques

- Étude complète de la turbine : une bonne partie du travail a d'abord consisté à comprendre comment fonctionnait vraiment la turbine. L'idée était de voir ce qu'elle pouvait produire comme tension, comment elle réagissait quand on la mettait en charge, et jusqu'où elle pouvait aller en courant. Ce qui a permis de voir ses limites et de savoir si elle pouvait suffire à alimenter le système.
- Optimisation du système de charge : ensuite, il a fallu s'assurer que l'énergie produite par la turbine pouvait bien être stockée. Plusieurs composants ont été testés pour filtrer et stabiliser le courant. Des diodes de protection ont été ajoutées, et le module de charge (le BQ25173) a été réglé pour correspondre aux vraies capacités du supercondensateur utilisé, valeurs qui pourront être utilisées de façon définitive sur la carte finale.
- Mesure de débit : un premier montage pour lire le débit d'eau a été mis en place. Comme la turbine génère un signal à chaque passage d'eau, il a fallu créer des circuits pour capter et traiter ce signal proprement. Plusieurs essais ont été faits pour obtenir quelque chose de stable et d'exploitable.
- Développement d'un prototype logiciel sous Arduino : une première version du programme a été écrite sur Arduino. Elle permet de lire le débit, calculer la consommation d'eau, mesurer la tension de la super capa, et d'afficher tout ça sur un petit écran e-paper.
- Mise en service de l'écran e-paper : l'écran e-paper a été testé et intégré dans le système. Même s'il n'est pas simple à utiliser, un affichage stable a pu être obtenue avec la carte NodeMCU. La phase de test a permis de bien comprendre comment il fonctionnait et de valider son intégration.
- Réalisation de tests en environnement simulé : un montage complet a été mis en place avec une pompe, des bassines et un débitmètre de référence pour simuler une vraie circulation d'eau. Ce qui a permis de tester tout le système dans des conditions proches du réel et de vérifier le bon fonctionnement de ce qui a été réalisé.

## 2. Contributions structurelles

- Adaptation du code sur NXP (MCUXpresso) : un début de portage du programme Arduino vers l'environnement NXP a été commencé. Bien que non abouti le travail commencé a été documenté avec des schémas et des explications facilitant la prise en main de la prochaine personne qui travaillera sur le sujet.
- Choix et commande des composants clés : une recherche a été menée pour sélectionner les composants adaptés au projet : microcontrôleur NXP, supercondensateurs plus fiables, diodes de protection, modules radio... Le tout en tenant compte des contraintes

techniques, mais aussi de la disponibilité sur le long terme (pour éviter les problèmes de stock ou d'obsolescence plus tard).

- Schémas de câblage clairs et documentés : des schémas ont été réalisés pour synthétiser les montages testés. L'objectif était de garder une trace propre du travail fait, que ce soit pour soi-même ou pour permettre à quelqu'un d'autre de reprendre facilement le projet après.
- Conception d'un premier PCB : Une carte électronique complète a été conçue pour réunir tous les composants testés jusque-là.

### 3. Apport à l'équipe

- Aide sur des tâches annexes au projet : en parallèle du projet principal, mon soutien a été sollicité sur diverses petites missions comme de la soudure, la fabrication de câbles, de l'aide sur du matériel à préparer. Des tâches simples mais qui ont permis de faire avancer certaines tâches plus vite et de soulager un peu l'équipe.
- Documentation claire et complète : Tout le travail a été structuré et documenté que cela soit les montages, les schémas, les résultats de tests, les choix de composants... L'idée était de laisser une trace exploitable, pour qu'un autre membre de l'équipe puisse facilement reprendre le projet sans devoir tout redécouvrir.

### 4. Bilan général

Le projet qui n'avait quasiment pas démarré est passé de simples mesures perdues dans d'anciens documents à quelque chose de bien plus structuré :

- Un prototype fonctionnel capable de mesurer un débit, d'estimer une consommation et d'afficher les données.
- Une première base de code amorcée sur NXP, avec quelques problématiques techniques identifiées.
- Un premier PCB commencé avec des instructions claires pour la suite.

Tout ce travail a permis de poser des bases solides, même si certains éléments ne sont pas encore terminés, de bonnes fondations ont été posées. Même si certains éléments restent à finaliser, le projet est maintenant bien lancé, avec une direction claire pour la suite.

## Prolongement du projet

À la fin du stage, le projet était encore en phase de développement et n'avait pas atteint son terme. Il restait encore plusieurs étapes importantes pour atteindre ne serait-ce qu'un prototype pleinement fonctionnel.

Tout d'abord, l'ensemble des tests prévus n'a pas pu être mené à bien, en particulier ceux impliquant les nouveaux composants commandés qui ne sont pas arrivés avant la fin du stage. Ces derniers auraient permis d'optimiser le système sous divers aspects (rendement énergétique, fiabilité des mesures, qualité du signal, etc.). Le schéma actuel pourrait donc être modifier par la suite.

Le circuit imprimé (PCB), bien qu'en cours de conception, n'était pas encore finalisé ni optimisé. Il reste à finaliser le schéma (la modélisation de certains composants n'étant pas terminée), effectuer des modifications, à lancer la commande de fabrication des PCB, puis à réceptionner les cartes pour pouvoir réaliser des tests.

Le développement du code sur la plateforme NXP n'a pas non plus été achevé. De nombreuses fonctionnalités restent à adapter ou à réécrire. L'adaptation depuis le code Arduino n'a pu être menée à terme, seuls les premiers essais liés à l'écran e-paper ont été commencés. Les fonctions principales du projet, comme la mesure de fréquence, le calcul du débit ou encore la lecture de la tension de la supercapacité, n'ont pas encore été intégrées. Seul l'initialisation avait été commencée, le code sur MCUXpresso n'est jamais allé au-delà de la couche bas niveau.

Les entrées/sorties, en particulier celles associées à des boutons ou capteurs externes, n'ont pas encore été implémentées ni testées.

La partie communication radio via antenne, destinée à transmettre les données à distance, n'a pas encore été abordée. Cette fonctionnalité est pourtant essentielle pour l'exploitation en conditions réelles du système, notamment dans des contextes de suivi à distance ou de relevé automatisé. Cependant l'antenne fait partie des composants commandés qui n'ont pas été reçus.

Enfin, le boîtier destiné à accueillir l'ensemble des composants doit encore être conçu et réalisé. Ce boîtier devra répondre à des contraintes comme étanchéité, la taille, l'accessibilité des connecteurs, etc. Une fois l'assemblage terminé, des tests en conditions réelles devront être menés afin d'évaluer la robustesse et la fiabilité de l'ensemble du système, puis procéder à d'éventuelles optimisations.

En l'état, le projet peut être estimé à environ 35 % de son avancement total. Il constitue ainsi une base solide pour un prolongement de projet.

## Apport personnel

Ce stage m'a énormément apporté, aussi bien sur le plan professionnel que personnel. J'ai été confronté à un projet concret, un sujet d'ingénieur, avec de vrais enjeux, un réel gros projet important pour l'entreprise qui m'a fait confiance et m'a accueilli ce qui m'a forcé à revoir ma manière de travailler. Dès les premières semaines, j'ai compris qu'il n'était pas possible d'avancer efficacement sans bases solides. Il m'est arrivé de penser qu'un test ou une vérification n'était pas nécessaire sur le moment pour finalement devoir revenir dessus plus tard, ce qui m'a fait perdre du temps. Avec l'expérience, j'ai appris à anticiper, à mieux structurer mon travail et à gagner en rigueur.

J'ai aussi réalisé qu'il n'était pas possible de travailler en connaissant « à peu près » un sujet. C'est quelque chose qui s'est démarquée notamment avec l'écran e-paper où je pensais l'avoir compris parce qu'il fonctionnait sur Arduino, mais en passant sur le microcontrôleur NXP j'ai réalisé qu'en réalité je ne comprenais pas réellement son fonctionnement. J'ai donc dû reprendre les choses à zéro, décortiquer son comportement, lire sa datasheet. Ce n'est que lorsque j'ai commencé à comprendre exactement comment fonctionnait le composant dans le moindre détail que j'ai pu commencer à réellement travailler dessus. Il est important de connaître parfaitement les composants ou outils qu'on utilise, surtout dans des cas comme celui-ci avec de la programmation bas niveau.

Ce stage m'a aussi appris à organiser mon travail. Notamment en programmation où j'ai rapidement réalisé que se plonger tête baissée dans le programme sans structure, ne fonctionne pas. J'ai dû apprendre à avancer étape par étape, à planifier ce que je faisais, et à prendre du recul pour identifier les priorités.

Une autre chose que m'a apporté ce stage était lors de la recherche de composants. Ce n'est pas juste aller chercher une référence sur Internet, il faut bien comprendre les contraintes du projet, savoir lire une datasheet, comparer plusieurs modèles, vérifier leur disponibilité à long terme, leur fiabilité et imaginer comment ils vont s'intégrer au reste du système.

Au-delà de l'aspect technique, ce stage m'a permis de découvrir le fonctionnement d'une petite entreprise privée. Chacun avance sur son propre projet de manière autonome tout en gardant la possibilité de demander de l'aide aux autres. Cette façon de travailler m'a permis de gagner en responsabilité, en rigueur et en autonomie.

## Conclusion

Ce stage de 14 semaines au sein de l'entreprise MicroBE avait pour objectif de concevoir un système autonome de mesure de consommation d'eau, ce qui a conduit à la problématique suivante « comment réaliser une carte électronique de mesure de consommation d'eau autonome ? ». À mon arrivée, le projet était quasiment au point mort avec seulement quelques tests non exploitables réalisés. L'objectif, dans l'idéal, aurait été d'aller jusqu'au bout du projet et de livrer un système totalement fonctionnel. Mais plus concrètement, il s'agissait surtout de poser des bases solides, de déblayer le terrain autant que possible, et de faire en sorte que les personnes qui reprendront le projet par la suite puissent s'appuyer sur un travail structuré, avec une bonne partie du développement déjà réalisée.

Le projet n'a effectivement pas pu être terminé, cependant plusieurs étapes clefs ont pu être réalisées comme l'étude de la turbine, la mise en place d'un système de mesure de débit et le développement d'un prototype logiciel sous Arduino. D'autres aspects, comme la conception du PCB ou le portage du code vers la plateforme NXP, ont été amorcés et posent déjà les fondations pour la suite du développement.

J'ai aussi découvert un environnement de travail très différent de ce que je connaissais. C'était une première pour moi de travailler dans une structure aussi petite, avec ce mode de fonctionnement. Il y a beaucoup d'autonomie et de liberté dans la manière de gérer son travail, ce qui est assez formateur, mais s'accompagne également de certaines limites, notamment au niveau du matériel ou du budget. L'utilisation de composants peu documentés ou difficiles à remplacer a parfois ralenti l'avancée du projet. Malgré tout, l'expérience a été très enrichissante et m'a énormément apporté.

J'ai particulièrement apprécié le travail de recherche, le fait d'être le premier à explorer une piste, à tester des idées, et surtout à devoir comprendre l'origine d'un problème. Devoir décortiquer tout le système, creuser chaque partie et chercher précisément d'où pouvait provenir l'erreur puis la satisfaction d'enfin résoudre le dysfonctionnement et voir le système fonctionner correctement.

Ce stage m'a conforté dans mon envie de poursuivre vers un métier d'ingénieur plutôt que de me limiter au métier de technicien.

## Lexique

### Diode TVS (Transient Voltage Suppression) :

Composant de protection utilisé pour absorber des pics de tensions transitoires dans un circuit. Elle protège le système en se déclenchant lorsque la tension dépasse un certain seuil.

### Diode Zener :

Diode conçue pour laisser passer le courant en sens inverse à partir d'une certaine tension dite "tension Zener". Elle est souvent utilisée pour réguler la tension dans un circuit ou comme protection contre les surtensions.

### Picoscope :

Oscilloscope numérique connecté à un ordinateur via port USB. C'est un outil compact très utile pour réaliser des captures d'écran directement sur l'ordinateur.

### Écran e-paper :

C'est un écran qui fonctionne avec de « l'encre électronique », le principe repose sur de minuscules capsules contenant des particules chargées électriquement, les particules blanches sont chargées positivement, les particules noires sont chargées négativement. Une fois l'écran mis à jour l'affichage reste même débranché, ainsi l'écran ne consomme de l'électricité que lorsqu'il se met à jour.

### NXP :

Entreprise de semi-conducteurs spécialisée dans la fabrication de microcontrôleurs, de processeurs et de circuits intégrés. Les microcontrôleurs utilisés dans ce projet proviennent de leur gamme MCX.

### Programmation bas niveau :

C'est la couche de programmation la plus basse, c'est-à-dire la couche la plus proche du microcontrôleur, cela consiste à contrôler directement le matériel, sans passer par des bibliothèques simplifiées. Elle demande de configurer soi-même les broches, les interfaces, les timers, etc. Cela offre plus de contrôle, mais nécessite de bien comprendre le fonctionnement du microcontrôleur.

### DesignSpark PCB :

Logiciel gratuit de conception de circuits imprimés (PCB). Il permet de créer des schémas électroniques, de faire le routage des cartes et de générer les fichiers nécessaires à la fabrication d'un circuit.

Supercondensateur (ou supercapacité) :

Composant de stockage d'énergie capable de se charger et de se décharger rapidement.

SPI (Serial Peripheral Interface) :

Protocole de communication série utilisé pour faire dialoguer un microcontrôleur avec des périphériques (écran, capteurs...). Repose sur plusieurs lignes : MOSI, MISO, SCLK, CS.

Référence RS :

Code unique attribué à un composant vendu par le distributeur RS Components. Il permet d'identifier facilement un produit dans leur catalogue, retrouver rapidement la fiche technique, les caractéristiques ou commander à nouveau un composant précis.



## Table des annexes

Annexe 1 planning prévisionnel initial

Annexe 2 Ancien schéma de câblage du système de mesure

Annexe 3 Mesure de test à vide

Annexe 3 (suite) Mesure de test supercapacité chargée

Annexe 3 (suite) Mesure de test supercapacité en charge

Annexe 4 comparaison des tests dans les 3 conditions

Annexe 4 (bis) Résultat des premières modifications

Annexe 5 résultat de test en modifiant un tuyau d'eau

Annexe 6 Résultats finales avec la diode TVS

Annexe 7 datasheet non officielle

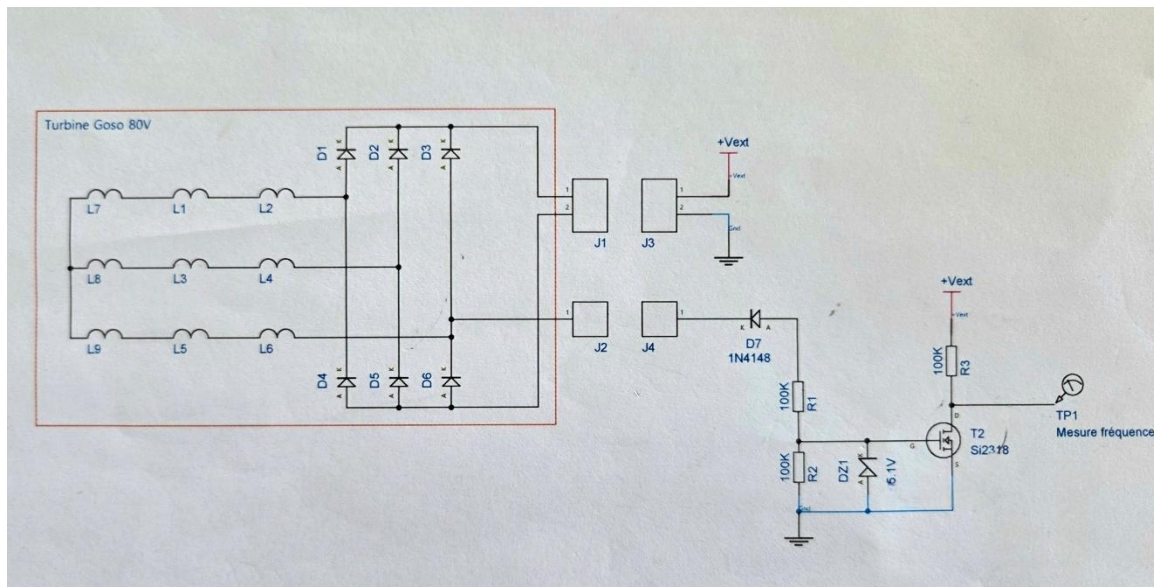
Annexe 8 interface de configuration des pins sur MCUXpresso

## Annexes

	semaine 1	semaine 2	semaine 3	semaine 4	semaine 5	semaine 6	semaine 7
mise en place du système de turbine							
alimentation de la super capa							
utilisation de la turbine comme un débitmètre							
mise en place système E/S							
affichage des données sur l'écran							
émission des données par radio							
vérification consommation							
vérification consommation système complet							

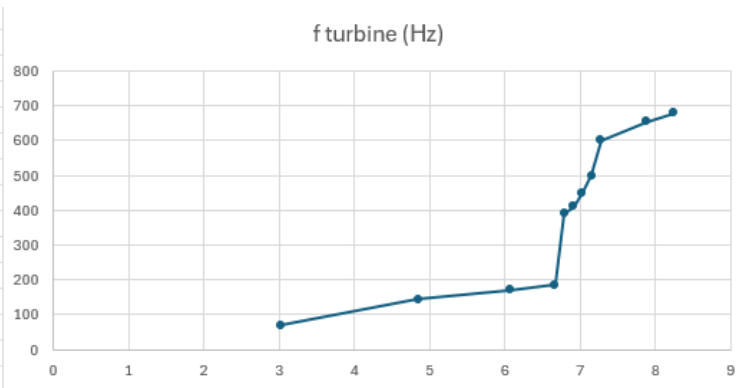
[illegible]

## Annexe 1 planning prévisionnel initial



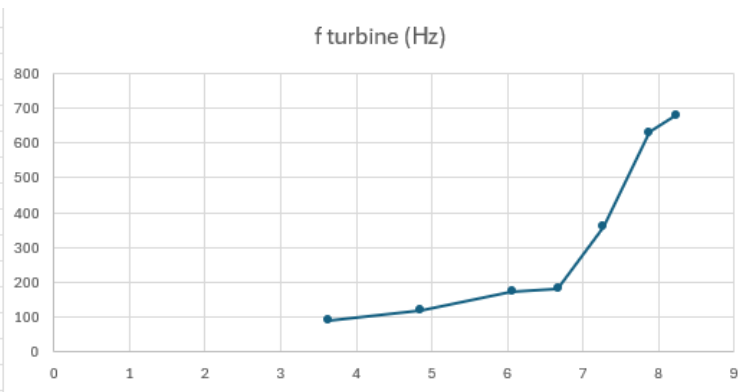
## Annexe 2 Ancien schéma de câblage du système de mesure

débit (L/min)	f débit mètre (Hz)	f turbine (Hz)
3,03030303	25	70
4,848484848	40	145
6,060606061	50	172
6,666666667	55	185
6,787878788	56	390
6,909090909	57	410
7,03030303	58	450
7,151515152	59	500
7,272727273	60	600
7,878787879	65	655
8,242424242	68	680



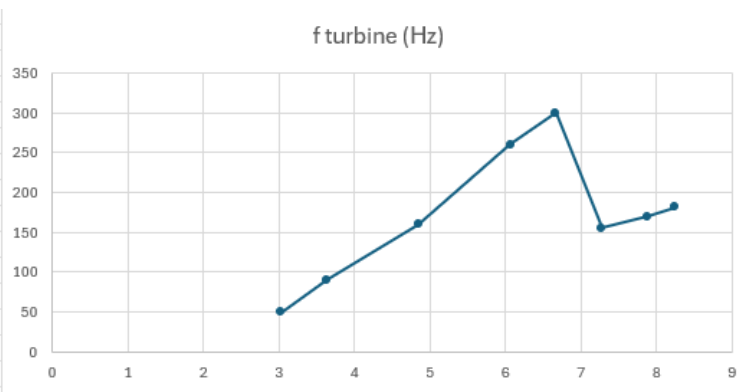
Annexe 3 Mesure de test à vide

débit (L/min)	f débit mètre (Hz)	f turbine (Hz)
3,636363636	30	92
4,848484848	40	121
6,060606061	50	175
6,666666667	55	183
7,272727273	60	360
7,878787879	65	630
8,242424242	68	680



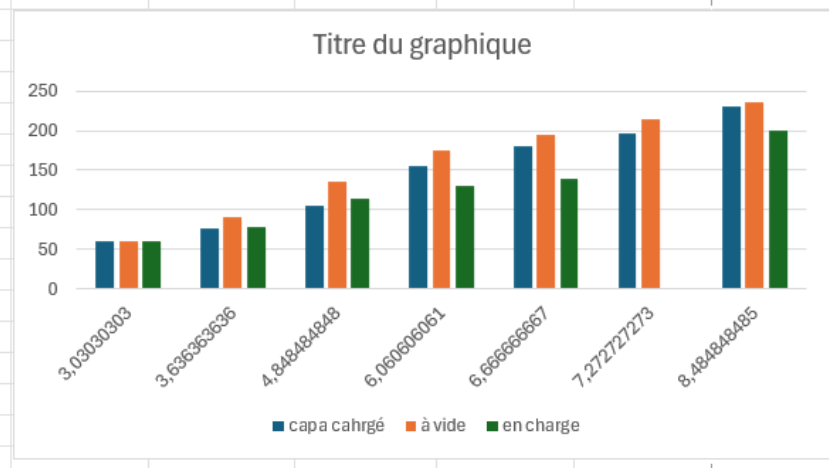
Annexe 3 (suite) Mesure de test supercapacité chargée

débit (L/min)	f débit mètre (Hz)	f turbine (Hz)
3,03030303	25	50
3,636363636	30	90
4,848484848	40	160
6,060606061	50	260
6,666666667	55	300
7,272727273	60	155
7,878787879	65	170
8,242424242	68	182

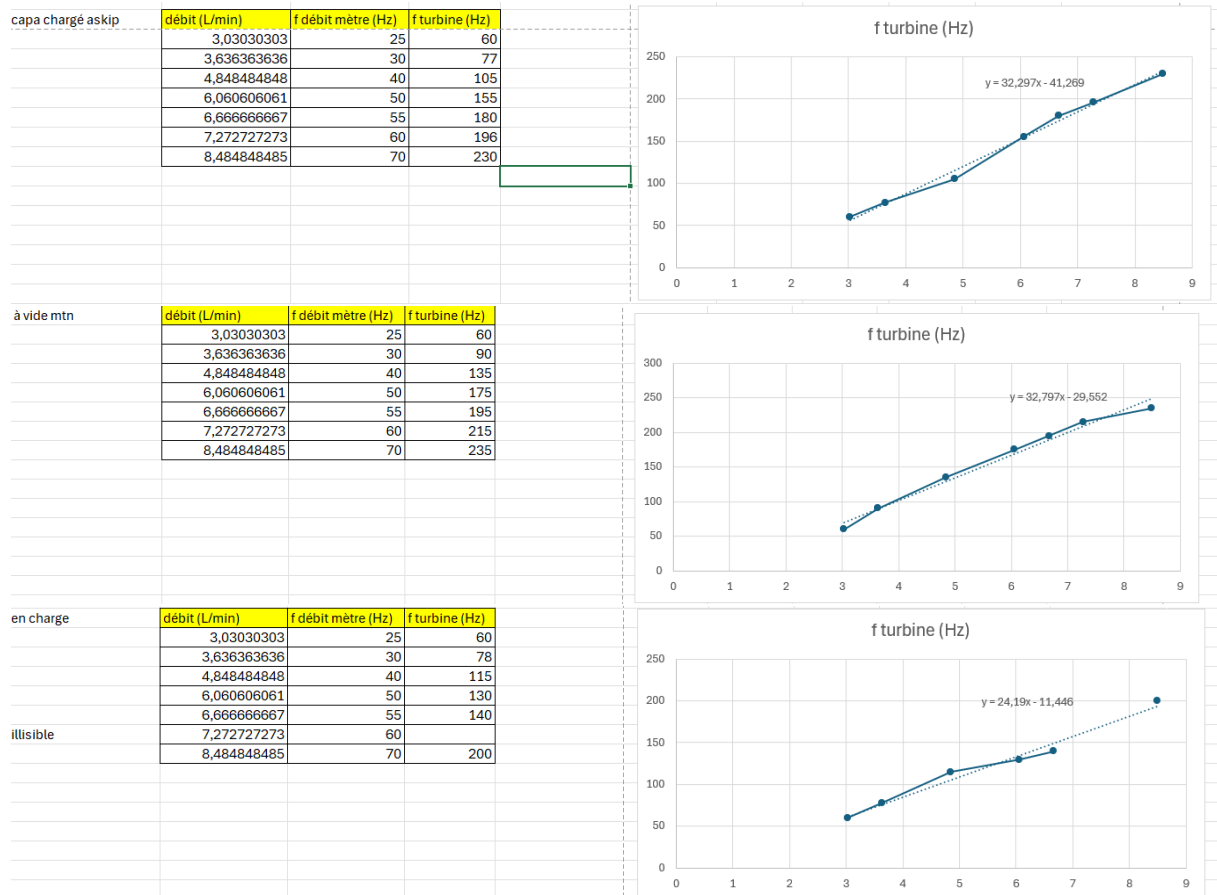


Annexe 3 (suite) Mesure de test supercapacité en charge

		fréquence de la turbine en Hz		
f débit mètre	débit (L/min)	capa cahrgé	à vide	en charge
25	3,03030303	60	60	60
30	3,63636364	77	90	78
40	4,84848485	105	135	115
50	6,06060606	155	175	130
55	6,66666667	180	195	140
60	7,27272727	196	215	
70	8,48484848	230	235	200

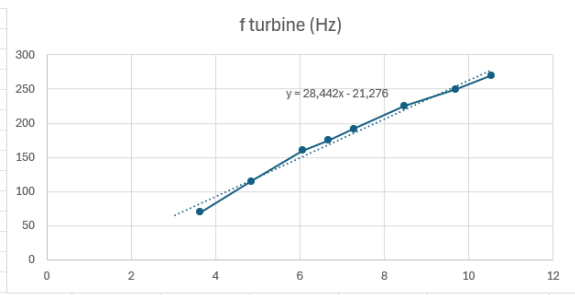


Annexe 4 comparaison des tests dans les 3 conditions



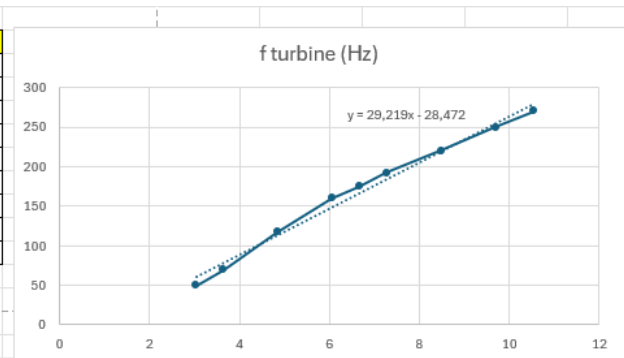
Annexe 4 (bis) Résultat des premières modifications

sans le tuyau capa chargé	débit (L/min)	f débit mètre (Hz)	f turbine (Hz)	tension turbine (V)
	3,03030303	25		
	3,636363636	30	70	2,6
	4,848484848	40	115	4,6
	6,060606061	50	160	6,4
	6,666666667	55	175	7
	7,272727273	60	192	7,6
	8,484848485	70	225	7,8
	9,696969697	80	250	8
	10,54545455	87	270	8,3

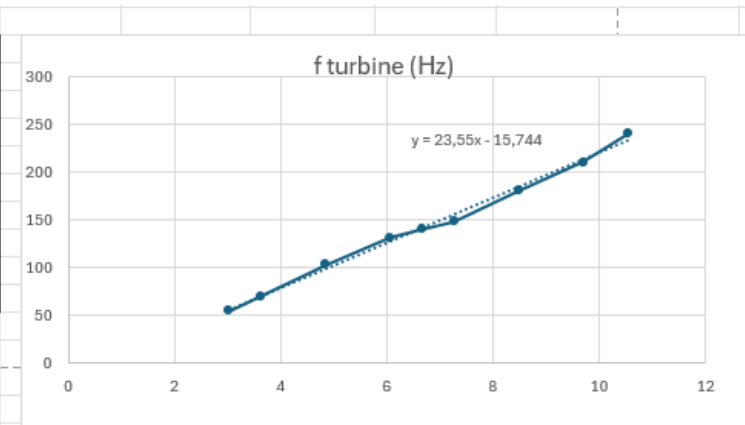


Annexe 5 résultat de test en modifiant un tuyau d'eau

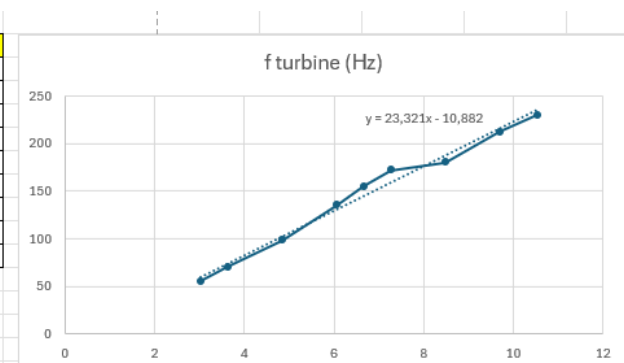
capa 2 chargé nouvelles mesures avec le système résistance + transistor	débit (L/min)	f débit mètre (Hz)	f turbine (Hz)
	3,03030303	25	50
	3,636363636	30	70
	4,848484848	40	117
	6,060606061	50	160
	6,666666667	55	175
	7,272727273	60	192
	8,484848485	70	220
	9,696969697	80	250
	10,54545455	87	270

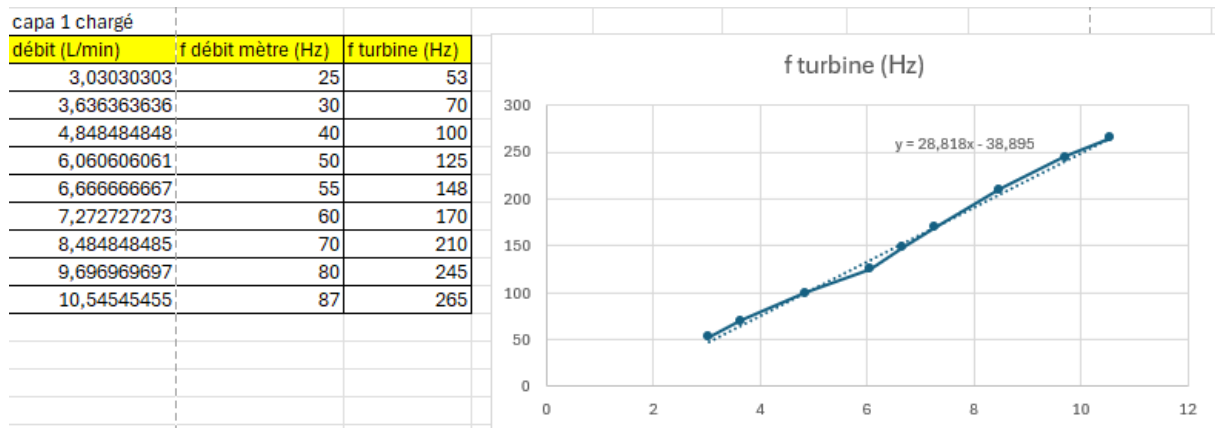


capa 2 en charge	débit (L/min)	f débit mètre (Hz)	f turbine (Hz)
	3,03030303	25	55
	3,636363636	30	70
	4,848484848	40	103
	6,060606061	50	131
	6,666666667	55	140
	7,272727273	60	148
	8,484848485	70	180
	9,696969697	80	210
	10,54545455	87	240



capa 1 capa en charge	débit (L/min)	f débit mètre (Hz)	f turbine (Hz)
	3,03030303	25	55
	3,636363636	30	70
	4,848484848	40	98
	6,060606061	50	135
	6,666666667	55	155
	7,272727273	60	172
	8,484848485	70	180
	9,696969697	80	212
	10,54545455	87	230





Annexe 6 Résultats finales avec la diode TVS

## Mise en œuvre d'un afficheur E\_paper



Afficheur MH-ET-LIVE	ESP8266 generic	Lolin NodeMCU V3 <sup>1</sup> CH340
Vcc	3,3v	3,3v
GND	GND	GND
SDI	GPIO13	D7
SCLK	GPIO14	D5
CS	GPIO0	D3
D/C	GPIO2	D4
RST	GPIO4	D2
BUSY	GPIO5	D1

## Solution basée sur la bibliothèque EPD supportant les afficheurs Waveshare

Page d'accueil de la bibliothèque : <https://github.com/soonuse/epd-library-arduino>

Téléchargement : <https://github.com/soonuse/epd-library-arduino.git>

Connexion électrique :

Paramétrage de la bibliothèque dans le fichier d'entête **epdif.h** :

```
#define BUSY_PIN    5 // D1
#define RST_PIN     4 // D2
#define DC_PIN      2 // D4
#define CS_PIN      0 // D3
```

Dans le code fourni, tous les fichiers source de la bibliothèque sont joint au projet Arduino pour figer l'application. Aucune installation n'est nécessaire.

Annexe 7 datasheet non officielle

## Solution basée sur les bibliothèques AdaFruit

bibliothèques à installer :

1. Adafruit Bus IO Library: [https://github.com/adafruit/Adafruit\\_BusIO](https://github.com/adafruit/Adafruit_BusIO)
2. Adafruit GFX Library : <https://github.com/adafruit/Adafruit-GFX-Library>
3. GxEPD : <https://github.com/ZinggJM/GxEPD>

la bibliothèque permet de gérer une multitude d'écrans, il faut donc choisir de dé-commenter la ligne correspondante à notre écran dans le code :

```
// select the display class to use, only one
// #include <GxDEPG0150BN/GxDEPG0150BN.h> // 1.50" b/w
// #include <GxDEPG0150C1/GxDEPG0150C1.h> // 1.54" b/w
// #include <GxGDEH0154D67/GxGDEH0154D67.h> // 1.54" b/w 200x200, SSD1681
// #include <GxGDEW0154T8/GxGDEW0154T8.h> // 1.54" b/w 152x152 UC8151 (IL0373)
// #include <GxGDEW0154M09/GxGDEW0154M09.h> // 1.54" b/w 200x200 JD79653A
// #include <GxGDEW0154M10/GxGDEW0154M10.h> // 1.54" b/w 152x152 UC8151D
// #include <GxGDEW0154Z04/GxGDEW0154Z04.h> // 1.54" b/w/r 200x200
// #include <GxGDEW0154Z17/GxGDEW0154Z17.h> // 1.54" b/w/r 152x152
// #include <GxGDEW0154Z90/GxGDEW0154Z90.h> // 1.54" b/w/r 200x200 SSD1681
// #include <GxGDEW0213I5F/GxGDEW0213I5F.h> // 2.13" b/w 104x212 flexible
// #include <GxGDE0213B1/GxGDE0213B1.h> // 2.13" b/w
// #include <GxGDEH0213B72/GxGDEH0213B72.h> // 2.13" b/w new panel
// #include <GxGDEH0213B73/GxGDEH0213B73.h> // 2.13" b/w newer panel
// #include <GxGDEW0213B74/GxGDEW0213B74.h> // 2.13" b/w 128x250 SSD1680
// #include <GxGDEW0213Z16/GxGDEW0213Z16.h> // 2.13" b/w/r
// #include <GxGDEH0213Z19/GxGDEH0213Z19.h> // 2.13" b/w/r UC8151D
// #include <GxGDEW0213T5D/GxGDEW0213T5D.h> // 2.13" b/w 104x212 UC8151D
// #include <GxDEPG0213BN/GxDEPG0213BN.h> // 2.13" b/w 128x250, SSD1680, TTGO T5 V2.4.1, V2.3.1
// #include <GxGDEH029A1/GxGDEH029A1.h> // 2.9" b/w
// #include <GxGDEW029T5/GxGDEW029T5.h> // 2.9" b/w UC8151 (IL0373)
// #include <GxGDEW029T5D/GxGDEW029T5D.h> // 2.9" b/w UC8151D
// #include <GxGDEW029T94/GxGDEW029T94.h> // 2.9" b/w
// #include <GxDEPG0290BS/GxDEPG0290BS.h> // 2.9" b/w Waveshare variant, TTGO T5 V2.4.1 2.9"
// #include <GxGDEW029Z10/GxGDEW029Z10.h> // 2.9" b/w/r
// #include <GxGDEH029Z13/GxGDEH029Z13.h> // 2.9" b/w/r UC8151D
// #include <GxGDEW026T0/GxGDEW026T0.h> // 2.6" b/w
// #include <GxDEPG0266BN/GxDEPG0266BN.h> // 2.66" b/w 152x296, SSD1680, TTGO T5 V2.66, TTGO T5 V2.4.1
// #include <GxGDEW027C44/GxGDEW027C44.h> // 2.7" b/w/r
// #include <GxGDEW027W3/GxGDEW027W3.h> // 2.7" b/w
// #include <GxGDEY027T91/GxGDEY027T91.h> // 2.7" b/w
// #include <GxGDEW0371W7/GxGDEW0371W7.h> // 3.7" b/w
// #include <GxGDEW042T2/GxGDEW042T2.h> // 4.2" b/w
// #include <GxGDEW042Z15/GxGDEW042Z15.h> // 4.2" b/w/r
// #include <GxGDEW0583T7/GxGDEW0583T7.h> // 5.83" b/w
// #include <GxGDEW075T8/GxGDEW075T8.h> // 7.5" b/w
// #include <GxGDEW075T7/GxGDEW075T7.h> // 7.5" b/w 800x480
// #include <GxGDEW075Z09/GxGDEW075Z09.h> // 7.5" b/w/r
// #include <GxGDEW075Z08/GxGDEW075Z08.h> // 7.5" b/w/r 800x480
```

Les écritures de texte peuvent utiliser plusieurs polices de caractère GFX, notamment celles contenues dans la bibliothèque AdaFruit, comme par exemple :

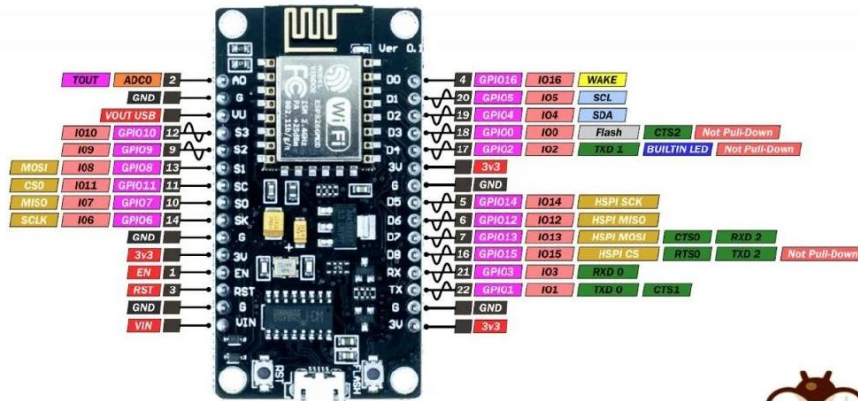
```
// FreeFonts from Adafruit_GFX
#include <Fonts/FreeMonoBold9pt7b.h>
#include <Fonts/FreeMonoBold12pt7b.h>
#include <Fonts/FreeMonoBold18pt7b.h>
#include <Fonts/FreeMonoBold24pt7b.h>
```

## Annexe : « pinout » des cartes utilisées pour le test

Annexe 7 (suite) datasheet non officielle

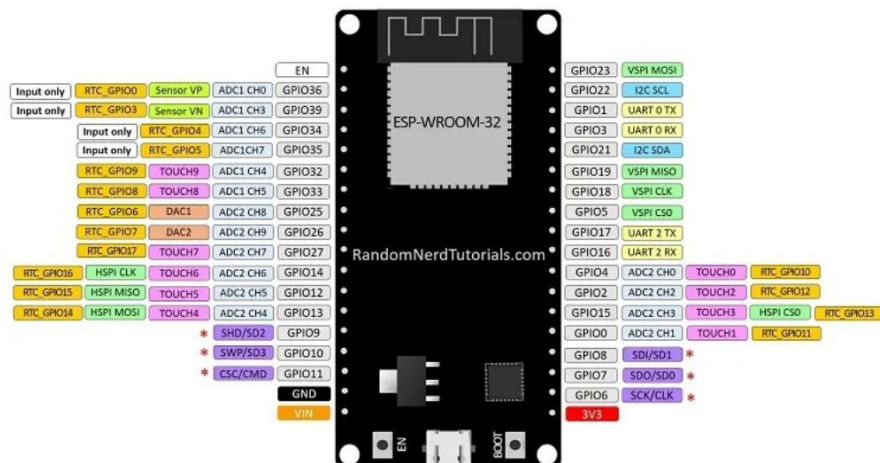


**NodeMCU v3 CH340** **PINOUT**



[www.mischianti.org](http://www.mischianti.org) 

# ESP32 DEVKIT V1 – DOIT



\* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

Annexe 7 (suite) datasheet non officielle

**Pins**

Pin	Pin name	Label	Identifier	Freedom Head	mikroBUS(TM)	GPIO	TPM
1	PTE0/CLKOUT...	J3[1]/UART1...	J3[1]	J5[4] (TX)	GPIOE-GPIO.0		
2	PTE1/SP11_M...	J3[3]/UART1...	J3[3]	J5[3] (RX)	GPIOE-GPIO.1		
3	VDD3	P3V3_KL27Z					
4	VSS4	GND					
5	USB0_DP	USB0_DP					
6	USB0_DM	USB0_DM					
7	USB_VDD	P3V3_KL27Z					
8	ADC0_DP1/A...	A0-ADC0_SE1	J4[2] (A0)	J6[1] (AN)	GPIOE-GPIO.16	TPM0-CLK	
9	ADC0_DP0/A...	A2-ADC0_SE0	J4[6] (A2)		GPIOE-GPIO.20	TPM1-CH...	
10	ADC0_DM0/A...	A3-ADC0_SE...	J4[8] (A3)		GPIOE-GPIO.21	TPM1-CH...	
11	ADC0_DP3/A...	THERMISTE...			GPIOE-GPIO.22	TPM2-CH...	
12	ADC0_DM3/A...	THERMISTE...			GPIOE-GPIO.23	TPM2-CH...	
13	VDDA	P3V3_KL27Z					
14	VREF_OUT	VREFH_C15					
15	VREFL	GND					
16	VSSA	GND					
17	CMPO_JN5/A...				GPIOE-GPIO.29	TPM0-CH...	
18	ADC0_SE23/A...	MIKROBUS_R...		J6[2] (RST)	GPIOE-GPIO.30	TPM0-CH...	
19	PTE3/TPM0...	D8-TPM0_CH4	J2[2] (DB)		GPIOE-GPIO.31	TPM0-CH...	

**Package**

MCXC242VLH - LOFP 64 package

**Routing Details**

#	Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	GPIO initial state	GPIO interrupt	Slew rate	Drive strength	Pull select
57	SPI0	PCS0	→	[57] SPI0_PCS0	J1[1]	SPI_CS_PCS	Output	n/a	n/a	Fast	Low	Pullup
58	SPI0	SCK	→	[58] SPI0_SCK	J1[3]/ACCEL_WAKE	SPI_SCLK_SCK	Output	n/a	n/a	Fast	Low	Pullup
28	GPIOA	GPIO_12	→	[28] PTA12	D2-TPM1_CH0	RESET_RST	Output	Logical 1	n/a	Slow	Low	Pullup
46	GPIOC	GPIO_3	→	[46] PTC3	J2[15]	DC	Output	Logical 1	n/a	Fast	Low	Pullup
27	GPIOA	GPIO_5	→	[27] PTA5	D9-TPM0_CH2	BUSY	Input	n/a	Interrupt/DMA request is disabled	Slow	Low	Pullup

**Problems**

Level	Resource	Issue
Warning	SPI0.spi_miso	Signal controller data in, ta...
Warning	LPUART0	Peripheral LPUART0 is not i...
Warning	USB0	Peripheral USB0 is not initia...

Annexe 8 interface de configuration des pins sur MCUXpresso